|  |
| --- |
| **机器学习** |
| **Otto Group Product Classification Challenge案例名** |
| 主 研 人：刘一阳  参 研 人： |
| 审 核 人：  方 向：工业算法案例研究  版 本 号：A |
|  |

**版本更新**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **更新描述** | **作者** |
| 2018/10/31 | A | 初稿 | 刘一阳 |

**选题表格**

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 竞赛名 | 竞赛背景描述（50字以内） | 类型（分类/回归） |
| 2018/10/31 | 奥托集团产品分类竞赛（Otto Group Product Classification Challenge） | 奥托集团是全球最大的电子商务公司之一，在20多个国家设有子公司，由于公司的规模庞大，因而对产品性能的一致分析至关重要。比赛要求选手根据产品的特性对产品进行分类。 | 分类 |

目录

[1. 背景描述 4](#_Toc25575_WPSOffice_Level1)

[1.1 竞赛赛题描述 4](#_Toc5940_WPSOffice_Level2)

[1.2 评估指标描述 5](#_Toc12357_WPSOffice_Level2)

[2. 数据来源及描述性统计分析 5](#_Toc5940_WPSOffice_Level1)

[2.1 大赛数据来源 5](#_Toc22683_WPSOffice_Level2)

[2.2 数据的描述性统计 5](#_Toc16174_WPSOffice_Level2)

[2.2.1 数据基本情况描述： 5](#_Toc5940_WPSOffice_Level3)

[2.2.2 数据字段介绍： 5](#_Toc12357_WPSOffice_Level3)

[2.2.3 数据描述性统计 6](#_Toc22683_WPSOffice_Level3)

[3. 优秀算法思路 6](#_Toc12357_WPSOffice_Level1)

[3.1 方案一 6](#_Toc32293_WPSOffice_Level2)

[3.1.1 方案一数据预处理及特征工程部分方案 6](#_Toc16174_WPSOffice_Level3)

[3.1.2 方案一模型设计、建立部分方案 6](#_Toc32293_WPSOffice_Level3)

[3.1.3 方案一结果、排名等 6](#_Toc29922_WPSOffice_Level3)

[3.1.4 方案一算法流程图 6](#_Toc4787_WPSOffice_Level3)

[3.2 方案二 7](#_Toc29922_WPSOffice_Level2)

[… 8](#_Toc4787_WPSOffice_Level2)

[4. 算法比较 8](#_Toc22683_WPSOffice_Level1)

[表4-1 算法比较 8](#_Toc7253_WPSOffice_Level2)

[5. 总结与展望 8](#_Toc16174_WPSOffice_Level1)

[5.1 总结 8](#_Toc23198_WPSOffice_Level2)

[5.2 建模思路 8](#_Toc16619_WPSOffice_Level2)

1. 背景描述

奥托集团（Otto Group）来自德国领先的电子商务解决方案及服务的提供商，在全球综合B2C排名中，仅次于亚马逊排在第二位，同时也是全球最大在线服装、服饰和生活用品零售渠道商 。

奥托集团（Otto Group）总部位于德国汉堡，是一家国际化集团，业务涉及零售、金融投资、物流服务等各大领域；奥托集团（Otto Group）是一个家族集团，她拥有强有力的、创新的零售品牌和其相关物流以及金融服务。截至2011年2月，已拥有49721名员工；在全球拥有123家公司；各子公司和分支机构遍布于欧洲、北美和亚洲的20个国家。

**1.1 竞赛赛题描述**

奥托集团是全球最大的电子商务公司之一，在20多个国家设有子公司，包括Crate＆Barrel（美国），Otto.de（德国）和3 Suisses（法国）。我们每天在全球销售数百万种产品，我们的产品线中添加了数千种产品。

对产品性能的一致分析至关重要。然而，由于我们多样化的全球基础设施，许多相同的产品被分类不同。因此，我们的产品分析质量在很大程度上取决于准确聚类类似产品的能力。分类越好，我们对产品系列产生的洞察就越多。

<https://www.kaggle.com/c/otto-group-product-classification-challenge#description>

**1.2 评估指标描述**

竞赛中评估模型优劣的指标。

选手的提交的评估指标是错层次对数损失。每个产品都被打上了一个真实的标签。对于每个产品，选手需要提交一系列预测的可能性（每个种类一个），衡量标准如下：

其中N为测试集中的产品数量，M为分类标签的数量，log为自然对数， 为1如果i属于类j，否则为0，是选手所预测的i属于类j的可能性。

选手所提交的对于某个产品的预测可能性的和不需要为1，因为他们在打分时会被重新划分（同一行的和做分母）。为了避免log的极端值，预测的可能性会被所替代.

1. 数据来源及描述性统计分析

**2.1 大赛数据来源**

描述大赛数据的来源

大赛数据提供方Otto Group，为大赛提供真实生产数据，从软硬件环境诸多方面提供大赛支撑。

这里是数据的超链接。

链接：https://pan.baidu.com/s/1sI4sYooOZ61GA3FMrXF2rw

提取码：lyno

**2.2 数据的描述性统计**

**2.2.1 数据基本情况描述：**

在本次比赛中，我们为超过200,000种产品提供了93个功能的数据集。目标是建立一个能够区分我们主要产品类别的预测模型。获奖模型将是开源的。

* + 1. **数据字段介绍：**

TrainData.csv - 训练集

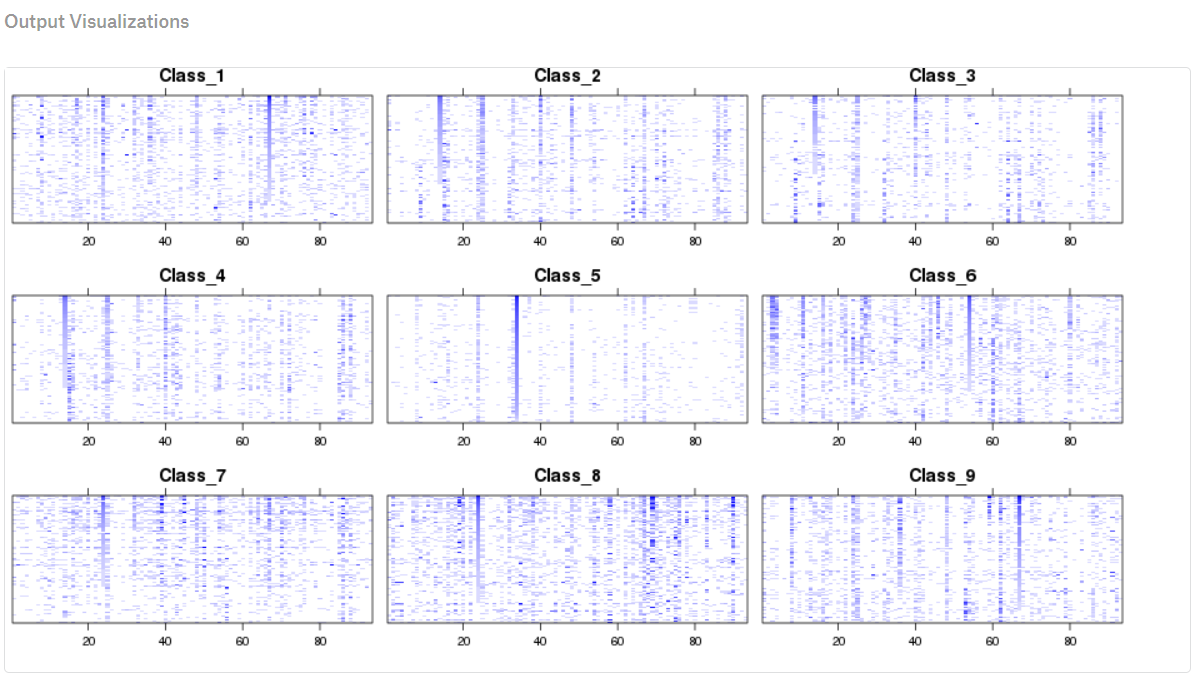
TestData.csv – 测试集

SampleSubmission.csv – 提交文件的样式

**表2-1 trainData.csv数据表字段介绍**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 变量含义 | 变量类型 | 缺失率 |
| **id** | **产品的id** | 名称 | 0% |
| **feat\_1, feat\_2, ..., feat\_93** | **产品的各项特性** | 离散 | 0% |
| **Target** | **产品的分类** | 离散 | 0% |

* + 1. **数据描述性统计**

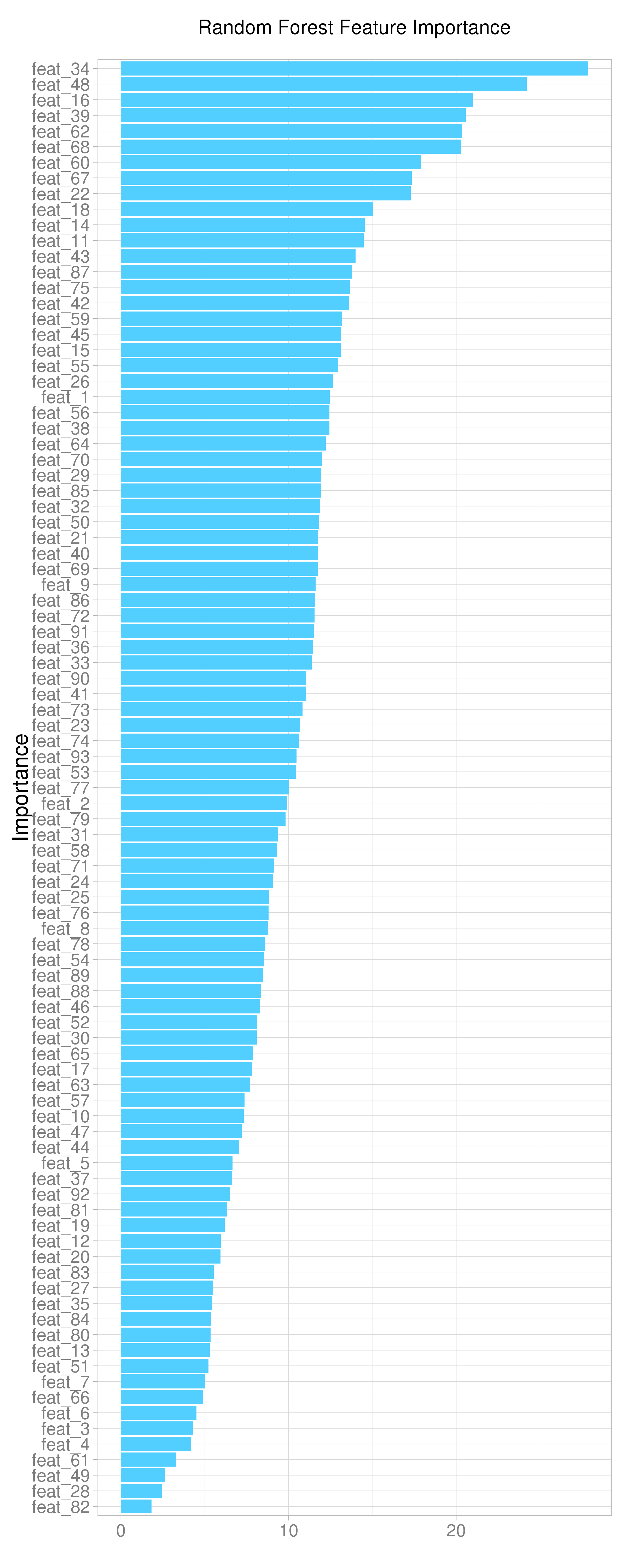


**图2-1 Visualization of the full dataset**

[**https://www.kaggle.com/herrahuu/visualization-of-the-full-dataset**](https://www.kaggle.com/herrahuu/visualization-of-the-full-dataset)



**图2-2 数据分析的代码实现**



**图2-3 随机森林分布**

3. 优秀算法思路

**3.1 方案一**

**3.1.1** 方案一数据预处理及特征工程部分方案

我们的解决方案基于3层学习架构，如附图所示。

第一级：大约有33个模型我们将他们的预测用作第二级的元特征，还有8个工程特征。

第二级：有3个模型使用33个元功能+第一级的7个功能训练：XGBOOST，神经网络（NN）和带有ExtraTrees的ADABOOST。

第三级：它由第二级预测的加权平均值组成。

第一层中的所有模型都使用5倍交叉验证技术进行训练，使用相同的折叠指数

**3.1.2** 方案一模型设计、建立部分方案

第二级的训练我们使用了4 Kfold random indices。它使我们能够在提交给排行榜之前计算得分。我们所有的交叉验证分数都与LB分数极为相关，因此我们对本地的绩效进行了很好的估计，这使我们能够为第二个学习级别丢弃无用的模型。

用于二级训练的模型和特征：

X =训练和测试集

一、 模型

1. RandomForest（R）.Dataset：X

2. Logistic回归（scikit）.Dataset：Log（X + 1）

3．额外树分类器（scikit）.Dataset：Log（X + 1）（但可能是原始的）

4．KNeighborsClassifier（scikit）.Dataset：Scale（Log（X + 1））

5．libfm.Dataset：稀疏（X）。每个特征值都是唯一的级别。

6．H2O NN。Bag of 10 runs.Dataset：sqrt（X + 3/8）

7．多项式朴素贝叶斯（scikit）.Dataset：Log（X + 1）

8．Lasagne NN（CPU）。Bag of 2 NN runs. 首先是Scale（Log（X + 1））数据集，其次是是数据集Scale（X）

9．Lasagne NN（CPU）.Bag of 6 NN runs.数据集：Scale（Log（X + 1））

10．T-sne。尺寸减小到3维。还使用T-sne 3尺寸堆叠2个kmeans功能。数据集：Log（X + 1）

11．Sofia（R）.Dataset：使用learner\_type =“logreg-pegasos”和loop\_type =“balanced-stochastic”对抗所有人。数据集：比例（X）

12．Sofia（R）。Trainned one against all with learner\_type="logreg-pegasos" and loop\_type="balanced-stochastic". Dataset: Scale（X，T-sne Dimension，基于randomForest important的13个最重要特征之间的一些3级交互）

13．Sofia（R）. Trainned one against all with learner\_type="logreg-pegasos" and loop\_type="combined-roc".数据集：Log（1 + X，T-sne Dimension，基于randomForest important的13个最重要特征之间的一些3级交互）

14．Xgboost（R）。Trainned one against all. 数据集：（X，按行划分的特征和（零））。用NA替换零。

15．Xgboost（R）。训练的Multiclass Soft-Prob。数据集：（X，7个Kmeans功能具有不同数量的簇，rowSums（X == 0），rowSums（Scale（X）> 0.5），rowSums（Scale（X）<-0.5））

16．Xgboost（R ）。训练的Multiclass Soft-Prob。数据集：（X，T-sne特征，X的一些Kmeans集群）

17．Xgboost（R）：训练的多类Soft-Prob。数据集：（X，T-sne特征，一些Kmeans集群的log（1 + X））

18．Xgboost（R）：训练的多类Soft-Prob。数据集：（X，T-sne特征，一些Kmeans Scale of Scale（X））

19．Lasagne NN（GPU）. 2-Layer. Bag of 120 NN runs with different number of epochs.

20．Lasagne NN（GPU）. 3-Layer. Bag of 120 NN runs with different number of epochs.

21．XGboost. 训练原始功能。Extremely bagged (30 times averaged).

22．KNN on features X + int（X == 0）

23．KNN on features X + int（X == 0）+ log（X + 1）

24．KNN on raw with 2 neighbors

25．KNN on raw with 4 neighbors

26．KNN on raw with 8 neighbors

27．KNN on raw with 16 neighbors

28．KNN on raw with 32 neighbors

29．KNN on raw with 64 neighbors

30．KNN on raw with 128 neighbors

31．KNN on raw with 256 neighbors

32．KNN on raw with 512 neighbors

33．KNN on raw with 1024 neighbors

二、 特征

1. 每个类的最近邻居的距离

2．每个类的2个最近邻居的距离之和

3. 每个类的4个最近邻居的距离之和

4. 每个类的最近邻居的距离在TFIDF空间中

5． T-SNE空间中每个类别的最近邻居的距离（3维）

6． 原始数据集的聚类特征

7． 每行中非零元素的数量

8． X （该功能仅用于NN二级培训）

第二级我们开始训练交叉验证只是为了选择最佳模型，调整超参数并找到最佳权重以平均第三级。

在我们找到一些好的参数后，我们使用整个训练集和袋装结果训练了第二级。

最终型号是一个非常稳定的第二级装袋：

XGBOOST：250次运行。

NN：600次运行。

ADABOOST：250次运行。

使用XGBOOST和NN的几何平均值，我们发现第3级的平均值更好。对于ET，我们做了一个先前结果的aritmetic mean：0.85 \* [ XGBOOST ^ 0.65 \* NN ^ 0.35] + 0.15 \* [ ET ]。

我们在第一级尝试了很多训练算法，如Vowpal Wabbit（许多配置），R glm，glmnet，scikit SVC，SVR，Ridge，SGD等......但这些都没有帮助提高二级性能。

我们还尝试了一些预处理，如PCA，ICA和FFT，但没有改进。

我们也尝试了功能选择而没有改进。似乎所有特征都具有正的预测能力。

我们也尝试了半监督学习而没有相关的改进，我们放弃它，因为它有很大的潜力来适应我们的结果。

定义解决此问题的最佳算法是：Xgboost，NN和KNN。T-sne减少也有很大帮助。其他算法对性能的参与较少。因此，我们学会了不丢弃低性能算法，因为它具有足够的预测能力来提高二级培训的性能。

**3.1.3** 方案一结果、排名等

最后结果：0.3962

排名：1/3514

**3.1.4** 方案一算法流程图

图片包含 杯子

描述已自动生成

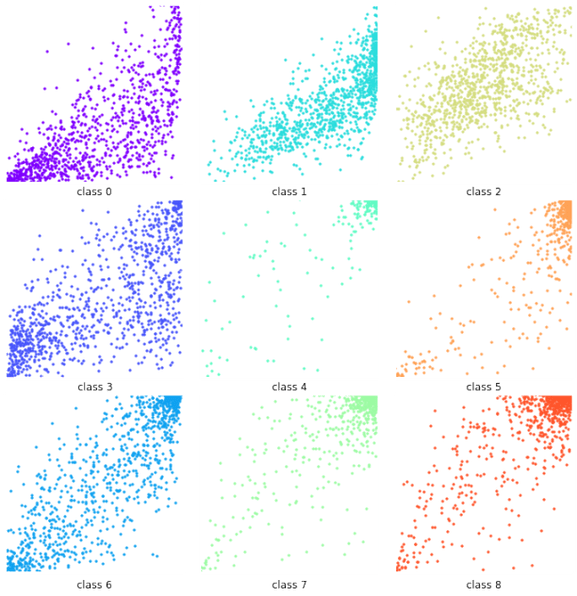
**图3-1**

**3.2 方案二**

**3.2.1** 方案二数据预处理及特征工程部分方案

在比赛开始时，我发现将数据分成两组是有用的：（1）训练和测试，（2）TF-IDF（训练）和TF-IDF（测试）

当涉及到集合时，你可以依赖的是你对算法的理解（基本上，你拥有的元素越多样越好），并努力尝试尽可能多的元特征。



**图3-2**

THE MORE DIVERSE METAFEATURES YOU HAVE, THE BETTER. METAFEATURE BY EXTRATREES VS METAFEATURE BY NEURAL NETWORK.

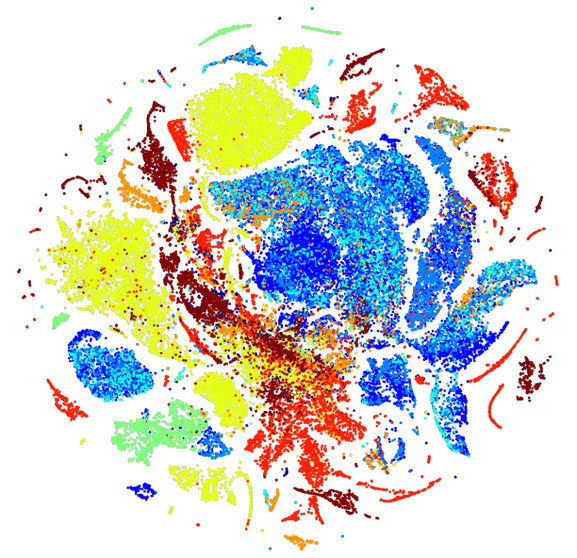
**3.2.2** 方案二模型设计、建立部分方案

我的解决方案的主要思想是堆叠。堆叠可以帮助您将不同方法的Y预测（或多标题问题时的标签）组合为“metafeatures”。基本上，为了获得火车的元特征，你将数据分成K个折叠，在K-1部分上训练K模型，同时预测每个K-1组留下的1个部分。要获得测试的元特征，您可以对这些K模型进行平均预测，或者根据所有列车数据进行单一预测。之后，如果您有多个元分类器，则可以在特征和元特征以及平均预测上训练元分类器。

谈到监督方法，我发现Xgboost和神经网络都能在数据上取得良好的效果。因此我决定在我的合奏中使用它们作为元分类器。

然而，KNN通常给出与决策树或神经网络非常不同的预测，因此我将它们作为元特征包含在第一级集合中。随机森林和xgboost也恰好用作元特征。

非常重要的是在第二级结合NN和XGB预测。虽然我的最终二级NN和XGB分别在私人LB上获得了大约.391，但他们的组合达到了.386，这是非常显着的改进。第二级套袋也有很大帮助。



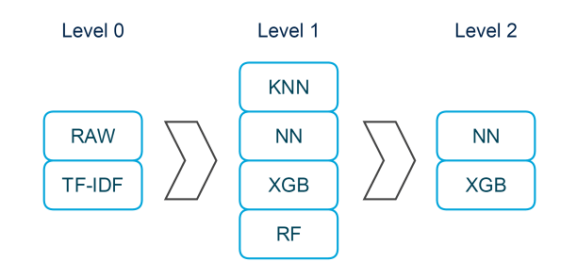
**图3-3 TSNE的2个维度**

**3.2.3** 方案二结果、排名等

结果：0.38655

排名：2/3514

**3.2.4** 方案二算法流程图



**图3-4 算法流程图**

4. 算法比较

**表4-1 算法比较**

|  | **评估指标** | **特征工程** | **基础算法** | **基本库** |
| --- | --- | --- | --- | --- |
| **算法1** | **0.38242** | **KNN, TFIDF space，NN** | **Xgboost, LASAGNE NN和ADABOOST ET** | **Sklearn, xsboost** |
| **算法2** | **0.38655** | **KNN, Random Forest, xhboost** | **NN,XGB,KNN,**  **Random forest** | **Sklearn,**  **xhboost, lasagne** |

制胜之处还是在参赛者的特征工程，排名第一的参赛者显然做了更加独特的特征工程，并且第一名利用权重进行计算，把XGBOOST,NN 和 ET三个模型的交叉验证预测集作为结果。

5. 总结与展望

**5.1 总结**

在查询过程中，我了解到这是kaggle最早也是最少欢迎的几个比赛之一，因而上面提供的两个方案能够在三千多支队伍中脱颖而出就显得格外不容易。对模型的堆叠是第二名的制胜法宝，而第一名的创新型则是自己引入了最后的带有权重的衡量计算公式。

在收集整理的过程中也学习到了这些优秀参赛者的一些方法和经验，为今后的学习和工作提供了宝贵的经验。

**5.2 建模思路**

在学习了两位选手的思路后，我无疑会采用堆叠的方式获取足够多的特征组合，并且还会采用多重的方式进行计算并最后引入权重进行计算从而获得更好的效果。

图片包含 杯子

描述已自动生成

**图5-1**