机器学习纳米学位

毕业项目——Gender Recognition by Voice

程铭 - 2017, 12, 19

I. 问题的定义

项目概述

在音频信号处理领域，利用音频的形式对性别进行识别目前存在一定的应用价值，比如在一些需要分辨男女身份的场合等。传统的判别方法大多是基于音频信号上的一些单一特性，如以男声的基音频率普遍较女声低来进行分类，分类方法相对单一，准确率低，该项目的出发点在于研究如何用机器学习的方法来利用多项音频特征建模，提升预测的准确率。本项目数据集选自 Kaggle 项目[https://www.kaggle.com/primaryobjects/voicegender]。

参考文献：

小波的提升方法在基音提取中的应用[J]. 彭辉,宁飞,孔宇. 山东大学学报(理学版). 2003(01)

根据语音分形维和基音周期的说话人性别识别研究[J]. 王振华. 生物医学工程学杂志 2008(04)

问题陈述

该问题是利用音频信号处理作为相关的领域知识，将原始音频信号用来提取部分信息作为可用特征的二分类问题。

利用音频信号处理之后提取的一系列特征参数，通过建立机器学习模型来实现上述问题的解决方案。

评估指标

在开始训练前将数据分为训练集，验证集和测试集三部分；在该身份识别中，选用准确率来衡量模型的表现：

$$准确率 = \frac{预测正确的数目}{测试数据总数} * 100\%$$

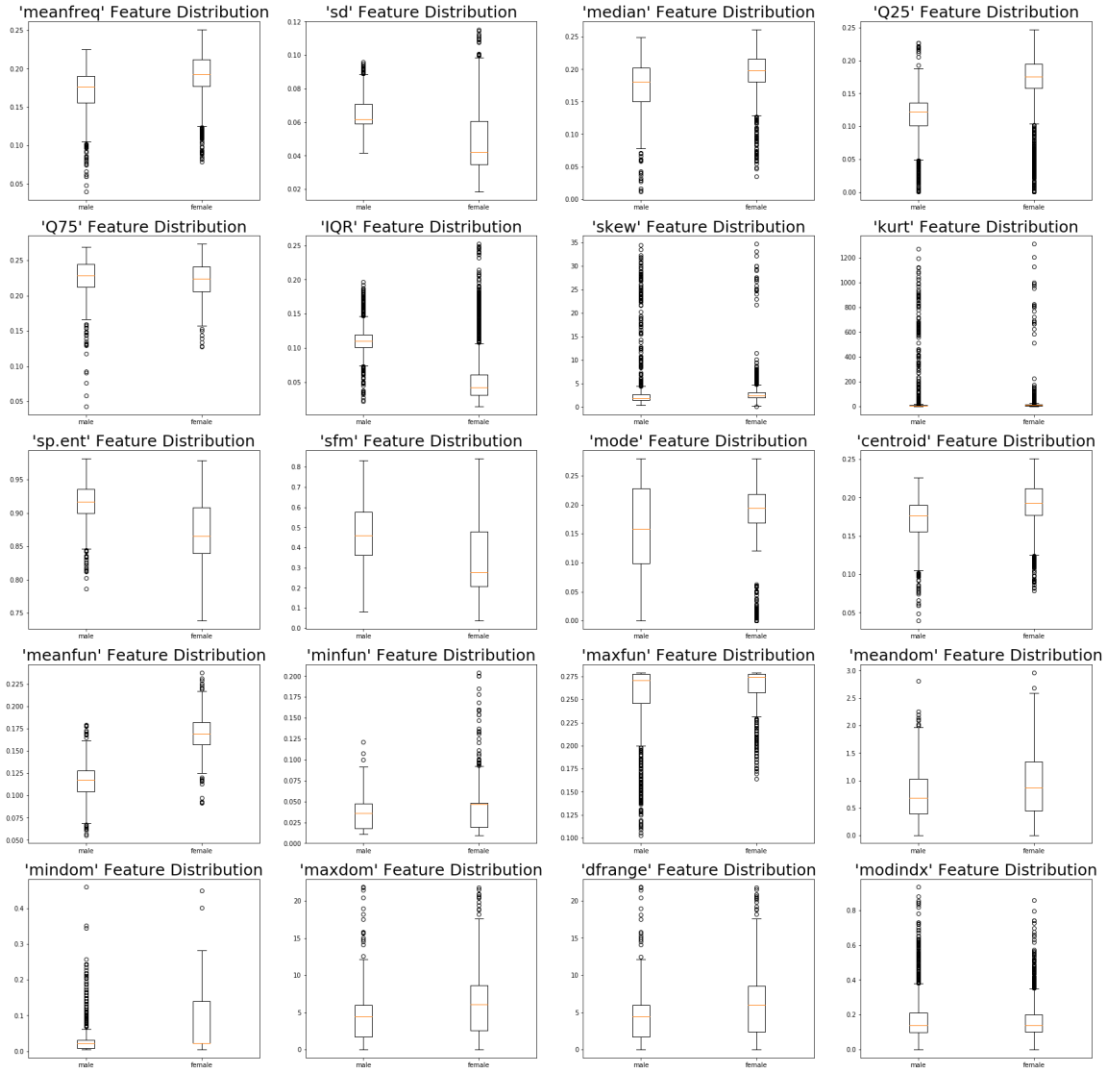
利用准确率作为评估标准来进行网格搜索，找出最优分类器参数；最后在测试集上进行预测，和在训练集上的准确率对比是否存在过拟合/欠拟合，计算准确率是否达到 98%以上，最终做出分类器模型是否合格的标准。

II. 分析

数据的探索

数据集由 Kaggle(https://www.kaggle.com/primaryobjects/voicegender)获得，其中一共包含了 3168 个样本，每个样本对应了 20 个经音频信号处理后提取的参数作为特征，及其相应的标签。特征包含频率平均值，频率标准差，频率中位数，频率第一四分位数，频率第三四分位数，频率四分位数间距，频谱偏度，频谱峰度，频谱熵，频谱平坦度，频率众数，频谱质心，峰值频率，平均基音频率，最小基音频率，最大基音频率，平均主频，最小最大主频，主频范围及累积相邻两帧绝对基频频差除以频率范围。

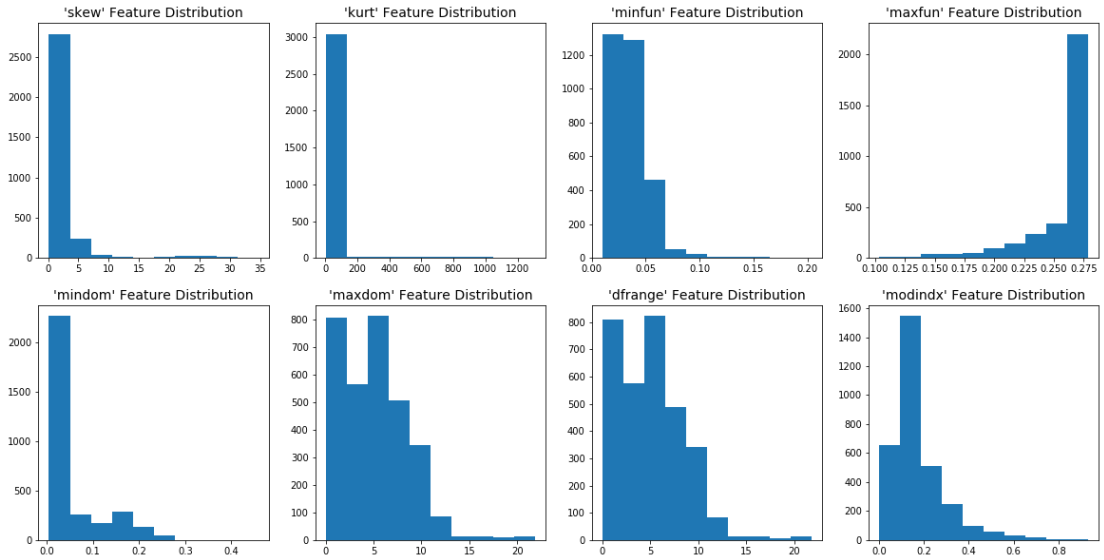
如下图 2-1 所示，将特征按箱子形图进行可视化，左侧为 ‘male’， 右侧为 ‘female’，其中比较明显的存在男女分布不同的特征有 meanfun, Q25, IQR，其余特征在 male 和 female 上箱形图的位置高低也有一定的区分度，因此这些特征可以用来有效地帮助解决分类问题。



2-1 特征箱形图可视化结果

探索性可视化

对所有特征可视化其分布情况，如下图 2-2 所示，发现以下 8 个特征存在明显的倾斜分布，需要在后续的处理中进行一定的数据预处理作为特征工程，以提升总体的模型效果。



2-2 存在倾斜分布的特征

经过如下箱形图 2-3 观察，可以发现 male 和 female 的不同类别，在 skew 和 kurt 两个特征上的分布并没有区别，且都偏侧化严重，经过后期的模型简单验证，发现确实删除这两项特征可以提高模型的表现，因此在此处将特征 ‘skew’ 及 ‘kurt’ 删除，不再使用；

2-3 无效特征示意图

算法和技术

整体步骤：

（1）在本毕业项目中，需要先利用提取出的如频率均值、频率标准差、频谱偏度、频谱峰度等音频信号处理的参数作为 20 个特征，对特征数据做一定的可视化后，根据观察结果对数据进行一定的预处理；

（2）采用集成学习中的 ‘随机森林（Random Forest）’ 算法，利用上述处理后的数据进行训练，根据验证集的准确率来选择相对最优，最适合这个模型的参数，再进一步在测试集上进行测试。

随机森林描述：

简称 RF，是 Bagging 的一个扩展变体；RF 在以决策树为基学习器构建 Bagging 集成的基础上，进一步在决策树的训练过程中引入了随机属性选择。具体来说，传统的决策树在选择划分属性时时在当前结点的属性集合中选择一个最优属性； 而在 RF 中，对基决策树的每个结点，先从该结点的属性集合中随机选择一个包含 k 个属性的子集，再从这个子集这种选出最优属性用于划分。
随机森林简单，容易实现，计算开销小，但可以在很多限时任务中展现出强大的性能。RF 中基学习器的多样性不仅来自样本扰动，还来自随机选择的属性扰动，这就使得最终集成的泛化性能通过个体学习器之间的差异度而进一步提升。

小结：
随机森林是一种基于集成学习的方法，训练时采用 booststrap 的取样方式，并利用了弱分类器的思想，可以很好的避免过拟合的情况发生，模型的泛化能力强；能根据训练来自动学习到不同特征的权重占比，对特征选择的要求不高；训练速度快，且适用本数据集情况下的高噪音的情况。

模型参数（部分）：
1. n_estimators: 随机森林中基决策树的数目
2. criterio: 决定利用信息增益或是基尼指数来计算划分属性结点的质量
3. max_features: 单棵树使用特征的最大数量
4. max_depth: 单棵树的最大深度
5. min_samples_split: 划分一个内部节点所需要的最小样本数目
6. min_samples_leaf: 在一个叶节点上所需要的最小的样本数目

算法流程：
1.样本集的选择——
每轮从原始数据集中有放回抽样（bagging）的方式抽取 N 个样例，共进行 k 轮抽取，即每轮抽取的训练集分别记为 T1，T2，T3，…，Tk。

2.决策树的生成——
假设特征空间共有 D 个特征，则在每一轮生成决策树的过程中，从 D 个特征中随机抽取选择其中的 d（d<D）个特征组成一个新特征集，通过使用新特征集来生成该决策树。

3. 模型的组合——
由于存在两个随机抽取的过程，即从原始数据中随机抽样和创建决策树时特征的随机选择，因此可以认为不同的基决策树之间应该是相互独立的，因此当将所有的决策树进行组合时，可以认为他们是同等重要，因此对所有基决策树的预测进行投票，即可得到该集成学习模型的预测结果。

参考资料来源： 《机器学习》-周志华

基准模型

基准模型来自 kaggle kernels - https://www.kaggle.com/nirajvermafcb/support-vector-machine-detail-analysis **Support Vector Machine detail analysis** 项目，该模型最后在验

证集上的准确率达到了 0.958990536278。本毕业项目选用随机森林，希望利用集成学习的思想来提高模型模型泛华能力，在测试集上获得 97%以上的准确率。

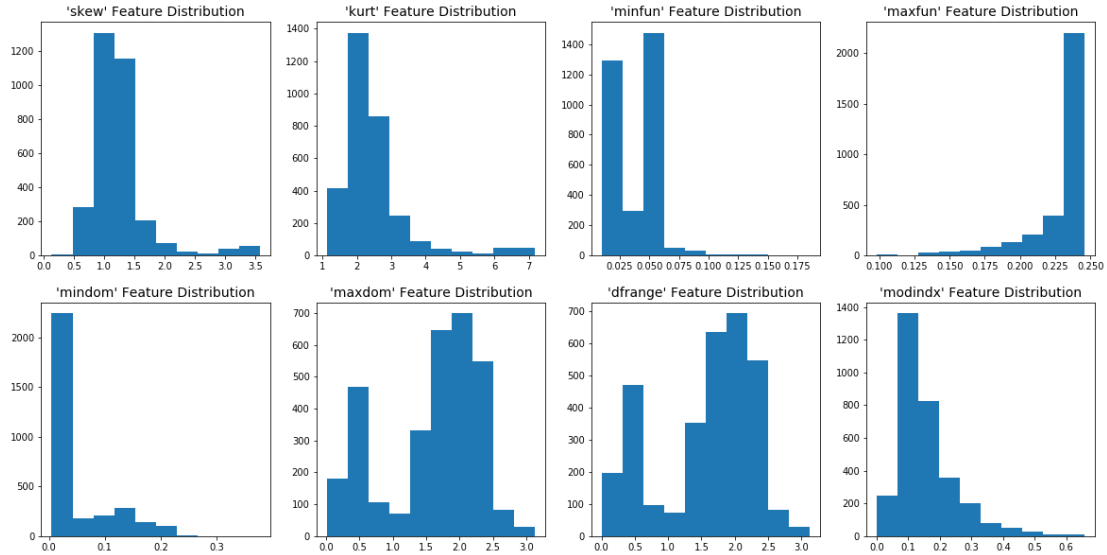
III. 方法

数据预处理

1、 利用 pandas 框架导入 csv 格式的数据，观察数据特征，一共包含 3168 个数据点，每个数据点有 20 个特征；

2、 将 label 一栏单独从原数据集中剥离，作为标签使用，同时对原数据格式 'male' / 'female' 数值化为 1/0；

3、 对不同特征的分布进行可视化，部分特征存在倾斜分布，如 *II.分析* 中图 2-2 所示。因此对特征'skew', 'kurt', 'minfun', 'maxfun', 'mindom', 'maxdom', 'dfrange', 'modindx'进行非线性变换（log），可以观察图 3-1，其倾斜分布有了一定程度的改善：



3-1 经过非线性变换后的特征分布

4、对所有特征进行归一化（I 利用 MinMaxScaler），以避免因数值大小的问题引起的权重不均衡；

5、经观察及尝试， 将特征 'skew'， 'kurt' 进行箱形图可视化，每张图的左侧为 male，右侧为 female，如下图 3-2 所示：

3-2 无效特征示意图

经过观察，可以发现 male 和 female 的不同类别，在该两个特征上的分布并没有区别，且都偏侧化严重，经过后期的模型简单验证，发现确实删除这两项特征可以提高模型的表现，因此在此处将特征 ‘skew’ 及 ‘kurt’ 删除，不再使用；

6、将原数据集分为训练集（80%）和测试集（20%），再将训练集中的 20%划分给验证集。

```
from sklearn.model_selection import train_test_split

#分离训练集，验证集，测试集
x_train, x_test, y_train, y_test = train_test_split(features, genders, test_size=0.2, random_state=40)
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size=0.2, random_state=40)

print("Training set has {} samples.".format(x_train.shape[0]))
print("Val set has {} samples.".format(x_val.shape[0]))
print("Testing set has {} samples.".format(x_test.shape[0]))
```

```
Training set has 2027 samples.
Val set has 507 samples.
Testing set has 634 samples.
```

执行过程

1. 创建一个训练和预测的流水线；需要从 scikit-learn 中导入部分项目所需要的库（评估函数和随机森林模型）。

Python 代码实现如下：

```
from sklearn.metrics import fbeta_score, accuracy_score
from time import time

def train_predict(learner, data_train, labels_train, data_test, labels_test):

    #训练分类器，并计算训练用时
    start_time = time()
    learner = learner.fit(data_train, labels_train)
    end_time = time()
    print('Training_time: ', end_time - start_time)

    #在训练集上预测，并打印准确率
    predictions_train = learner.predict(data_train)
    print('Accuracy_train:  ', accuracy_score(labels_train, predictions_train))

    #在测试集上预测，并打印准确率
    predictions_test = learner.predict(data_test)
    print('Accuracy_test:  ', accuracy_score(labels_test, predictions_test))

    return
```

2. 利用 scikit-learn 库中的随机森林模型进行建模（默认参数）

Python 代码实现如下：

```
from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(random_state = 20)

train_predict(learner=clf, data_train=x_train,
              labels_train=y_train, data_test=x_test, labels_test=y_test)
```

```
Training_time:  0.0621640682220459
Accuracy_train:   0.997039960533
Accuracy_test:   0.970031545741
```

完善

建立 GridSearchCV 网格搜索，对随机森林模型进行调参，找到最优模型，并通过在测试集

上的 Accuracy 来衡量最终表现。

具体步骤：
Step1 - 确定参数 n_estimators
经过网格搜索，对 n_estimators 从 2 取值到 100 进行搜索，得出该参数的最佳值为 80，此时模型在验证集上的准确率为 0.980276134122。

Step2 - 确定参数 criterion
经过网格搜索，对 criterion 从基尼指数和信息熵增益两方面比较，在上验证集表现两方法准确率基本相差不大，但 oob_score_（袋外得分）来说 gini 指数更高，因此可以判定其具有更好的泛化能力，选用 'gini'。

Step3 - 确定参数 max_features
经过网格搜索，单独对 max_features 按[auto, sqrt, log2, None]进行搜索，得出该参数的最佳值为 auto。

Step4 - 确定参数 max_depth
经过网格搜索，单独对 max_depth 进行搜索，得出该参数的最佳值为 14。

Step5 - 确定参数 min_samples_leaf&min_samples_split
经过网格搜索两参数，选用默认值即可，网格搜索并没有提高模型的性能。

小结：
经过对主要几个参数的网格搜索，模型在验证集上的准确率由"执行过程"中初步的预测值提升到了 0.980276134122，其性能提升主要是由参数 n_estimators 的调节而得到。

IV. 结果

模型的评价与验证

1. 经实际测试，模型训练时长 Training time：1.7366480827331543；且将测试集数据放入当前的最优模型中测试，得到该模型在测试集上的准确率为 0.983498349835。

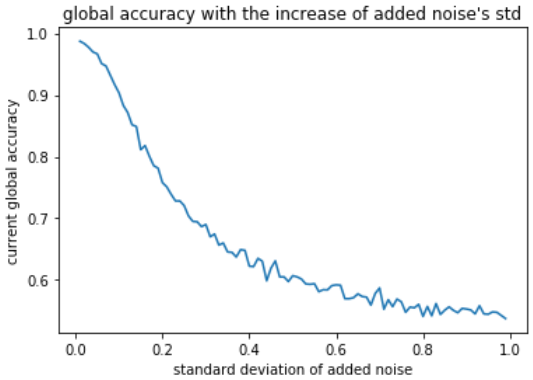
Python 代码实现如下：

```
#得到最优模型在测试集上的表现
predictions_test = best_clf.predict(x_test)

accuracy_test = accuracy_score(y_test, predictions_test)

print('Accuracy_test:   ', accuracy_test)
```

```
Accuracy_test:    0.981072555205
```

结论：模型已达到预期目的，评价指标准确率在验证集和测试集上均表现良好，达到了预先设定的 baseline；同时训练时间<2s，计算开销小，效果显著。

2. 利用 Numpy 库生成均值为 0，标准差从 0.01 到 1 的高斯噪声，人工加入全部原始数据后，放入模型进行准确率验证，得到全局准确率随添加的高斯噪声标准差增大而减小的趋势如图 4-1。



4-1 模型准确率随加入噪声标准差变化的趋势

结论：在小幅度的干扰下（高斯噪声的标准差<0.05），模型仍能维持在 97%以上的准确率；在较小幅度的干扰下（高斯噪声的标准差<0.1），该模型仍能维持在 90%以上的准确率，因此认为该模型鲁棒性良好。

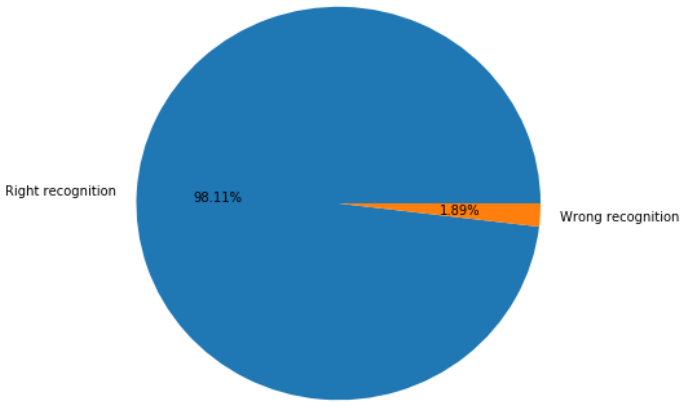
合理性分析

1. 根据前述基准模型，其在测试集上的准确为 0.958990536278。经对比，本文算法在测试集上的准确率已达到 98.11%，甚至在加入一定小幅度的高斯噪声下，准确率亦高于所选定基准模型，可以认定该模型较好地完成了既定任务。

2. 经过反复分析及尝试，因语音信号在不同个体之间差异较大，且较容易受到环境影响，经音频信号处理后的特征数据仍然存在较大干扰及异常点；基于上述，本项目的关键在于对音频信号特征进行合适的预处理工作。因此，经过对特征的可视化探索，及合理尝试，采用了III.方法中所述的数据预处理工作，并选用集成学习的经典模型：随机森林，最终成功解决了该项问题。

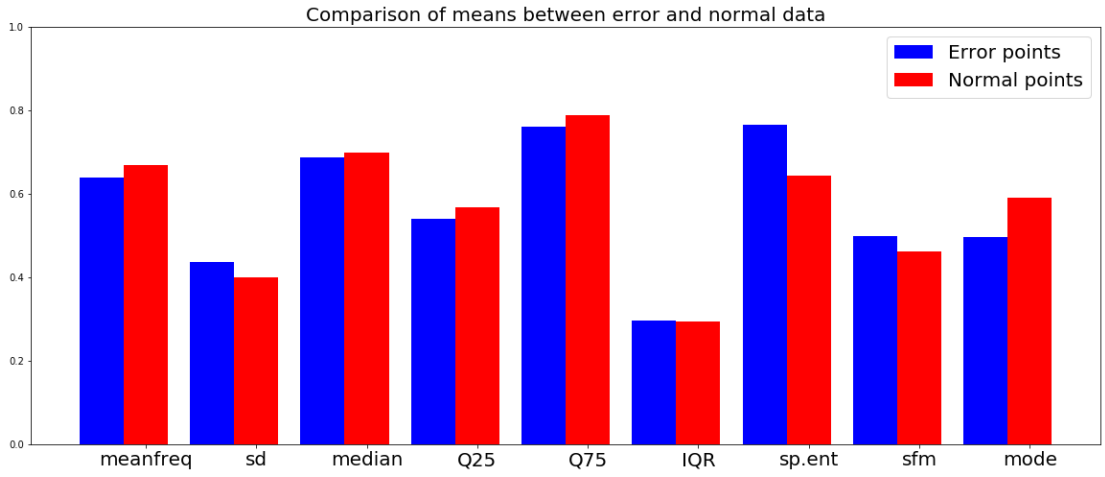
V. 项目结论

结果可视化

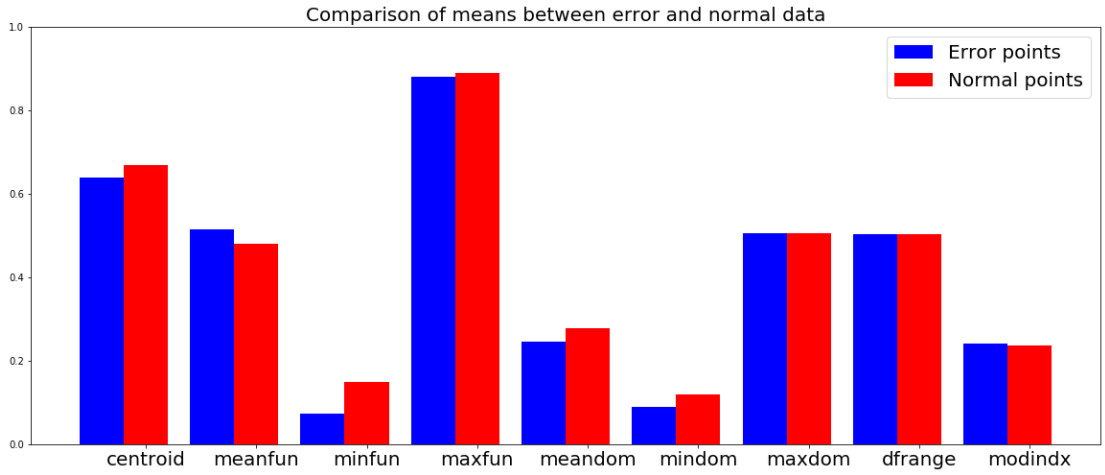


5 -1 最优模型在测试集上的表现

上图 5-1 中的测试集总数为 634 个数据点，其中分类正确 622 个，错误分类 12 个。将分类错误的 12 个数据点单独提出作为一个错误数据集，并求该集合所有特征的均值，后与全部数据集的数据作比较，可视化如图 5-2。



5-2（a）前 9 个特征的对比数据



5-2（b）后 9 个特征的对比数据

由上述可得结论：

分类错误的数据点中，普遍在 sp.ent，mode，minfun 共 3 特征上均值与正常数据的均值存在较大差异，因此初步判断为存在异常数据而对最终的预测产生了影响。

因语音信号提出的复杂性，易干扰性，在数据集中往往存在于个别数据与正常数据存在较大差异，由此可带来对算法的严重干扰，因此本项目中的核心工作主要在数据预处理部分进行了不断的分析与尝试，最后得到了在验证集上 0.981072555205 的准确率，可以认为效果显著，算法有效。

对项目的思考

在整个项目的完成过程中，一步步实现了从读取数据，数据预处理，模型选择，调参优化到结果可视化并做一定分析的完整的数据分析流程，在这个过程中也熟悉了 matplotlib，numpy，scikit-learn 等辅助工具的官方文档，无论是从分析问题的能力还是结局问题的效率上都获得了很大的提升。

项目的主要难点在于：
（1）没有经过系统的学习数据预处理的方法，从而在这个过程中经过了大量的尝试，不断与后置模型的准确率进行对比，以找出能更好地解决异常数据点的方法。
（2）通过对比不同的机器学习模型，经简单分析后逐个尝试，在模型选择上花费功夫较多，最后还是选择基于集成学习的方法来解决复杂问题；最后通过认真阅读 Random Forest 相关资料，理解各项参数的含义，以对调参工作加以优化。

模型最终的表现结果符合预期，达到了基本要求，且在一定的噪声干扰下仍能保持较好的准确率，可以认定该模型具有一定的鲁棒性。

需要作出的改进

本项目所实现所得的最终模型，在准确率以及鲁棒性上均得到了较好的结果，可以有效解决所提出的问题。若以本实验最终模型作为基准模型，应该还可以做进一步的提升，可以从以下几个方面加以考虑：
（1）在音频信号的处理方面，由 *II.分析-数据的探索* 部分箱图可以看出，处理所得的各项原始特征在男女不同类别上存在的差异并不大。因此，从特征选择的角度，应该从音频信号处理的角度研究更好的特征工程方法，获取在不同的类间方差更大的特征提取手段。
（2）在数据预处理方面，对异常数据的检测除了考虑 IQR 方面之外，也需要阅读更多的资料，以找到是否存在更好的异常值检测方方法，能够准确地处理数据。
（3）在模型选择方面，由实验数据来看最优模型在训练集上的准确率达到了 100%，高于测试集的水平，可以看出还是存在一定情况的过拟合问题，需进一步改进或更换模型（如 XGBoost），减少方差-变差之间的困境。