**摘 要**

本文使用了Kaggle上的公开数据集口袋妖怪数据，综合这门课程所学知识，对宝可梦的不同分类（普通宝可梦和传说宝可梦）影响因素进行探究，并得出相关结论。本文首先介绍了宝可梦的相关背景，其次对数据进行了一系列统计描述，展现变量的相关性和分布特点，以及普通宝可梦和传说宝可梦在各个特征上的异同；并对原始数据进行清洗；之后基于数据的特点，建立逻辑回归、SVM、决策树、随机森林、K近邻算法和朴素贝叶斯分类器等模型进行分类，接着采用集成算法对模型做出改进；最终找到影响宝可梦类别的关键因素。研究结果表明，特殊攻击、特殊防御和速度是识别普通宝可梦最关键的三个因素。

**关键词：**宝可梦；二分类；机器学习；集成算法；

目录

[1. 研究背景 4](#_Toc61376343)

[2. 数据介绍 4](#_Toc61376344)

[2.1. 数据来源 4](#_Toc61376345)

[2.2. 描述分析 5](#_Toc61376346)

[2.2.1. 总体概况 5](#_Toc61376347)

[2.2.2. 相关性分析 7](#_Toc61376348)

[2.2.3. 可视化 9](#_Toc61376349)

[2.2.3.1. 连续变量 9](#_Toc61376350)

[2.2.3.2. 分类变量 9](#_Toc61376351)

[2.2.3.3. 序数变量 10](#_Toc61376352)

[2.3. 数据清洗 11](#_Toc61376353)

[3. 模型选择 11](#_Toc61376354)

[3.1. 建立模型 11](#_Toc61376355)

[3.1.1. 逻辑回归模型 11](#_Toc61376356)

[3.1.2. SVM支持向量机 11](#_Toc61376357)

[3.1.3. 决策树 12](#_Toc61376358)

[3.1.4. 随机森林 12](#_Toc61376359)

[3.1.5. K近邻算法 12](#_Toc61376360)

[3.1.6. 朴素贝叶斯 13](#_Toc61376361)

[3.1.7. 精度对比 13](#_Toc61376362)

[3.2. 集成算法 15](#_Toc61376363)

[3.2.1. Voting Classifier 15](#_Toc61376364)

[3.2.2. Bagging 15](#_Toc61376365)

[3.2.3. Boosting 15](#_Toc61376366)

[3.2.3.1. AdaBoost 16](#_Toc61376367)

[3.2.3.2. Gradient Boosting 16](#_Toc61376368)

[3.2.3.3. XGBoost 16](#_Toc61376369)

[3.2.4. 精度对比 16](#_Toc61376370)

[3.3. 特征重要性 17](#_Toc61376371)

[4. 结论 18](#_Toc61376372)

## 研究背景

宝可梦是GAME FREAK开发的游戏及其系列动画中的生物。截止2020年，宝可梦（Pokemon）的世界已经经历了8个世代。全部的宝可梦加起来已经有893只，如果算上同一种宝可梦的不同形态，那么总数更是超过了千只。

宝可梦的神话是从游戏开始的（1996年2月赤/绿，1996年10月集换试卡牌），而后改编成动画和漫画（1997年4月），再有了剧场版（1998年7月）。而这一切的核心，就是893只宝可梦。

神兽也称为传说中的宝可梦。在宝可梦的世界中拥有独特的地位。神兽可以分为一级神兽、二级神兽和究极异兽，广义上还包括幻兽。其共通点在于神兽比一般的宝可梦要强大，且数量稀少。一般来说，他们都是有特殊的获取途径，无法大量的捕获，也无法通过生蛋来进行大量繁殖。截止第八世代，传说的宝可梦一共有59种。

## 数据介绍

### 数据来源

本文的数据来源于Kaggle公开数据集,共包含800个宝可梦的数据，主要包括三个方面：①主要属性：包括第一、第二属性；②基本属性：包括生命值、攻击、防御、特殊攻击、特殊防御、速度、总计和该宝可梦的世代；③该宝可梦是否为传说宝可梦。

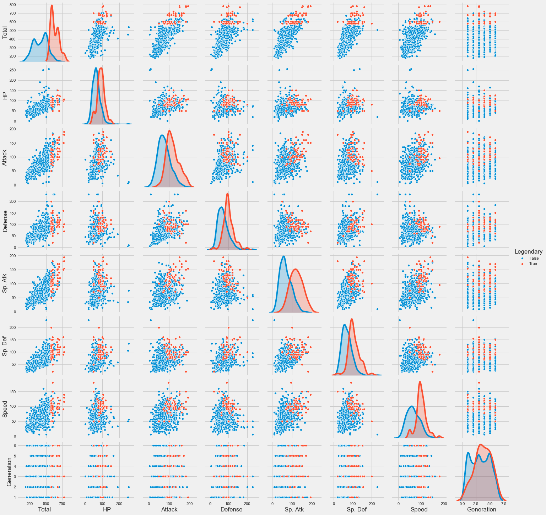
所有变量名称、介绍、类型及取值范围如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 变量名称 | 变量介绍 | 变量类型 | 取值范围 |
| 属性1 | 每个口袋妖怪都有一个属性，这决定了他们的克制与被克制关系 | 分类 | 1-18 |
| 属性2 | 还有一些口袋妖怪是双重属性的 | 分类 | 1-18 |
| 生命值 | 生命值，在被击倒之前宠物小精灵可以承受多少的伤害 | 数值 | 1-255 |
| 攻击 | 物理攻击的基础数值 | 数值 | 5-190 |
| 防御 | 抵抗物理攻击的基础数值 | 数值 | 5-230 |
| 特殊攻击 | 特殊攻击，特殊攻击的基础数值（例如火焰爆炸，泡沫射线） | 数值 | 10-194 |
| 特殊防御 | 抵抗特殊攻击的基础数值 | 数值 | 20-230 |
| 速度 | 决定宠物小精灵的进攻顺序 | 数值 | 5-180 |
| 总计 | 所有的统计数据的总和，某种程度上决定了精灵的强弱 | 数值 | 180-780 |
| 世代 | 该精灵在第几世代中出现 | 有序 | 1-6 |
| 传说 | 该精灵是否为传说 | 分类 | 0-1 |

### 描述分析

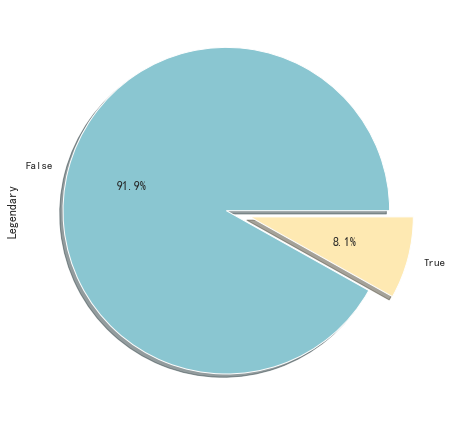
#### 总体概况

首先对各变量的分布情况和两两关系进行基础描述分析，如下图：



可以从经过分类后的相关中发现，不论是从对角线上的分布图还是从分类后的散点图，都可以看出对于传说宝可梦和普通宝可梦，其各基础属性的分布差异较大，故这些指标可以帮助我们去识别不同种类的宝可梦。

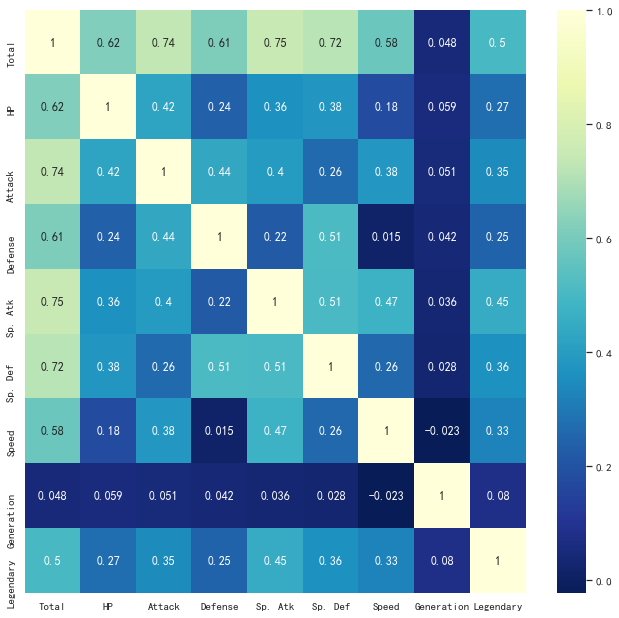
接着画出传说宝可梦和普通宝可梦占比的饼状图，传说宝可梦的占比仅有8.1%，说明数据分布不均衡。



#### 相关性分析

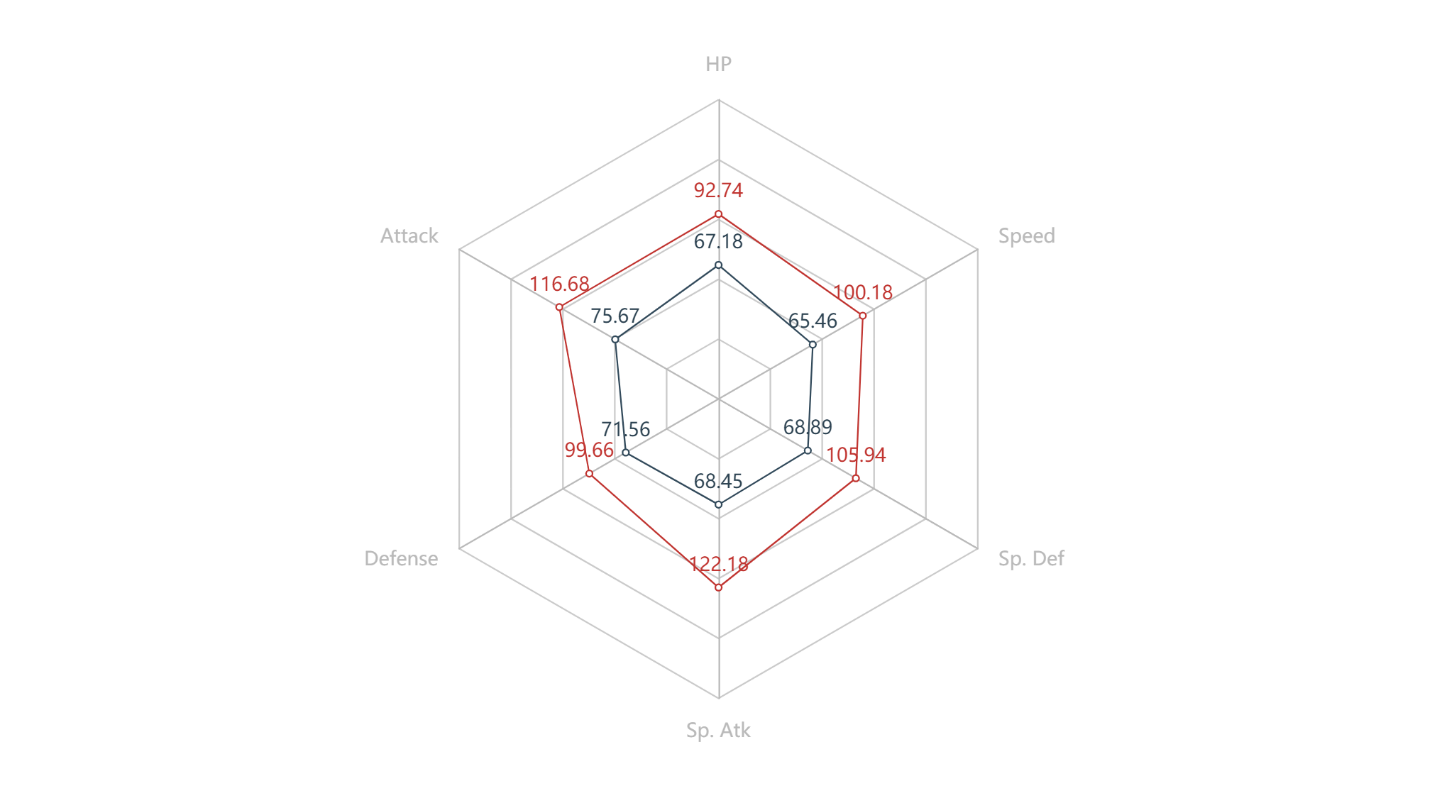
对各数值型变量初步进行相关系数的计算，初步探究指标之间的相关性，，为后续的模型使用以及变量选择提供参照。相关系数越靠近1，则说明变量之间的相关性越强。

将相关系数矩阵绘图如下。其中，格子上的数值代表其对应问题的相关系数，格子颜色越浅，表示变量相关性越高。从下图可以看出，大多数问题之间的相关系数都小于0.5，但总计（Total）和生命值、攻击、防御、特殊攻击、特殊防御、速度的相关系数都达到了0.5以上，在回归的时候需要对这些变量进行筛选，从而避免多重共线性带来的系数估计不准的问题。



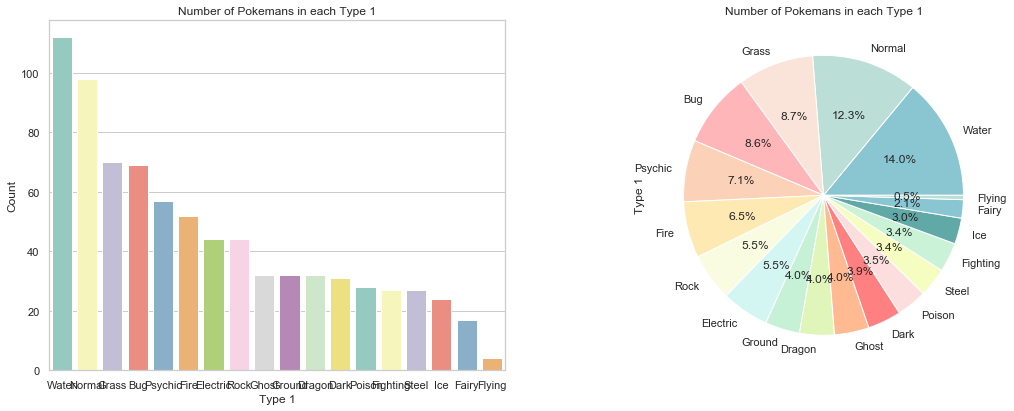
#### 可视化

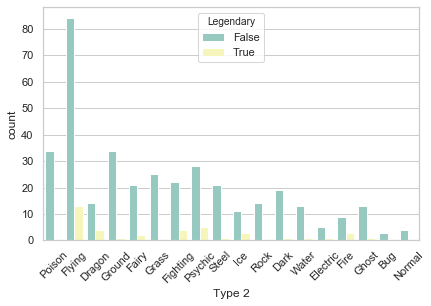
##### 连续变量

画出传说宝可梦和普通宝可梦生命值、攻击、防御、特殊攻击、特殊防御、速度属性均值的雷达图，发现在各项属性上传说宝可梦的数值均明显高于普通宝可梦，其中特殊攻击属性二者差距最大，传说宝可梦几乎是普通宝可梦的二倍。

##### 分类变量

统计各主属性（type1）的宝可梦的数量，并绘制条形图如下：

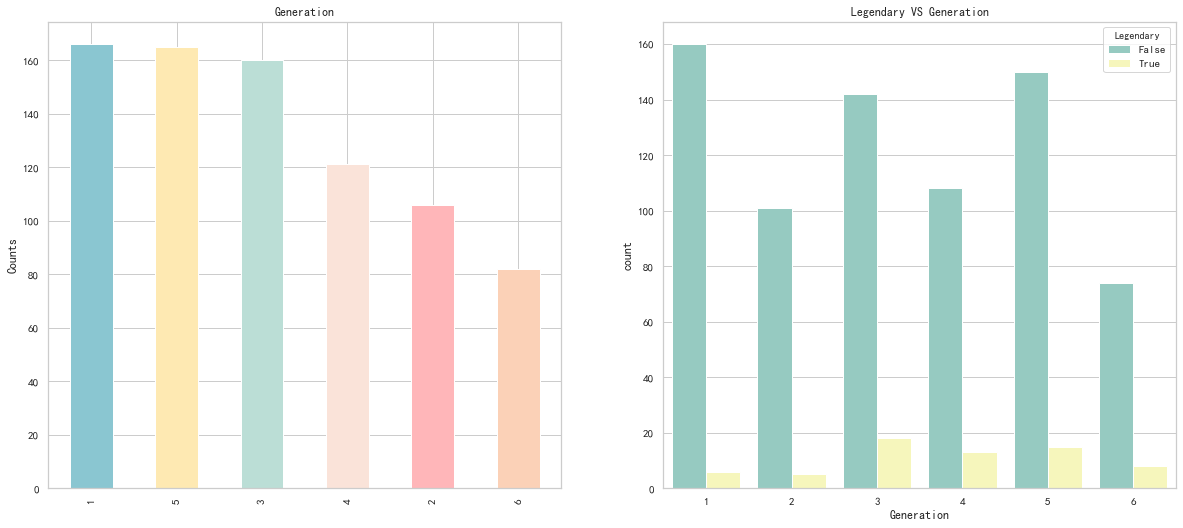




可以看到，各宝可梦的主属性中，水属性和普通属性较多，飞行属性的很少。从分类柱状图中，**可以粗略看出Dragon,flying和psychic是神兽的可能性较高。**

因为type2的缺失值较多，故这里不再做分析。

##### 序数变量



在第三、四、五代出现神兽的比例较高，分别为11%，11%，9%。

### 数据清洗

数据清洗是为了处理不合理的数据。主要包括：

1. 删除无意义的变量：宝可梦的编号和名字在分析和建模中没有用处，故删除这两个变量。
2. 删除冗余变量：基本属性中的总计（total）为生命值、攻击、防御、特殊攻击、特殊防御、速度之和，可以由它们完全表示，若保留会出现多重共线性，故将该变量剔除；
3. 缺失值处理：经过统计，属性2中存在大量缺失值（因为只有部分宝可梦精灵具有双重属性），难以进行插补，故直接删除该变量。
4. 数据类型转换：原始数据中Type1为字符串变量，需要将其转换为哑变量。

## 模型选择

在完成数据清洗后，采取不同模型进行回归，并进行比较和选择。

### 建立模型

#### 逻辑回归模型

logistic回归是一种广义线性回归（generalized linear model），实际上是分类模型，并常用于二分类。Logistic Regression 因其简单、可并行化、可解释强深受工业界喜爱。

逻辑回归在测试集上的准确率为0.9625。

#### SVM支持向量机

支持向量机（Support Vector Machine, SVM）是一类按监督学习（supervised learning）方式对数据进行二元分类的广义线性分类器（generalized linear classifier），其决策边界是对学习样本求解的最大边距超平面（maximum-margin hyperplane）。

SVM使用铰链损失函数（hinge loss）计算经验风险（empirical risk）并在求解系统中加入了正则化项以优化结构风险（structural risk），是一个具有稀疏性和稳健性的分类器[2]  。SVM可以通过核方法（kernel method）进行非线性分类，是常见的核学习（kernel learning）方法之一。

分别使用线性核函数SVM和高斯核函数SVM建立分类模型，得到测试集准确率分别为0.9625和0.95，结果相差不大。

#### 决策树

决策树是一个预测模型，他代表对象属性与对象值之间的一种映射关系。决策树是一种树形结构，其中每个内部节点表示一个属性上的测试，每个分支代表一个测试输出，每个叶节点代表一种类别，决策树是一种十分常用的分类方法。对于决策树，数据的准备往往是简单或者是不必要的，而且能够同时处理数据型和常规型属性，在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

决策树在测试集上的准确率为0.93125。

#### 随机森林

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，对于不平衡数据集来说，它可以平衡误差，还可以评估自变量的重要性。

随机森林在测试集上的准确率为0.95，相较于单棵决策树的精度略有提高。

#### K近邻算法

KNN是通过测量不同特征值之间的距离进行分类。它的思路是：如果一个样本在特征空间中的k个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别，其中K通常是不大于20的整数。KNN算法中，所选择的邻居都是已经正确分类的对象。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。和朴素贝叶斯之类的算法比，KNN对数据没有假设，准确度高，对异常点不敏感。但在样本不平衡的时候，对稀有类别的预测准确率会降低。

KNN在测试集上的准确率为0.95625.

#### 朴素贝叶斯

最为广泛的两种分类模型是决策树模型(Decision Tree Model)和朴素贝叶斯模型（Naive Bayesian Model，NBM）。和决策树模型相比，朴素贝叶斯分类器(Naive Bayes Classifier 或 NBC)发源于古典数学理论，有着坚实的数学基础，以及稳定的分类效率。同时，NBC模型所需估计的参数很少，对缺失数据不太敏感，算法也比较简单。理论上，NBC模型与其他分类方法相比具有最小的误差率。但是实际上并非总是如此，这是因为NBC模型假设属性之间相互独立，这个假设在实际应用中往往是不成立的，这给NBC模型的正确分类带来了一定影响。

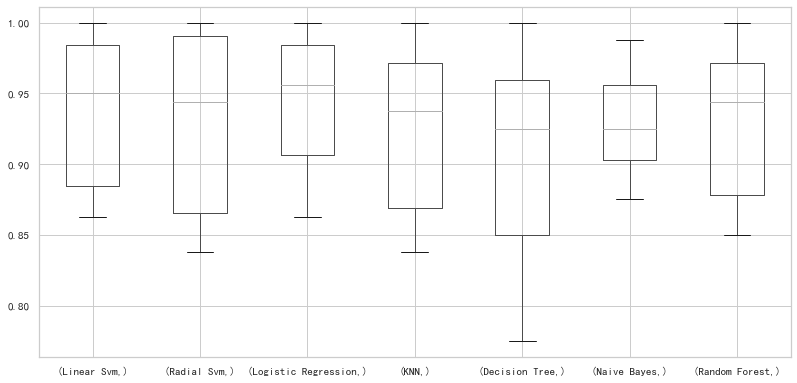
朴素贝叶斯在测试集上的准确率为0.94375.

#### 精度对比

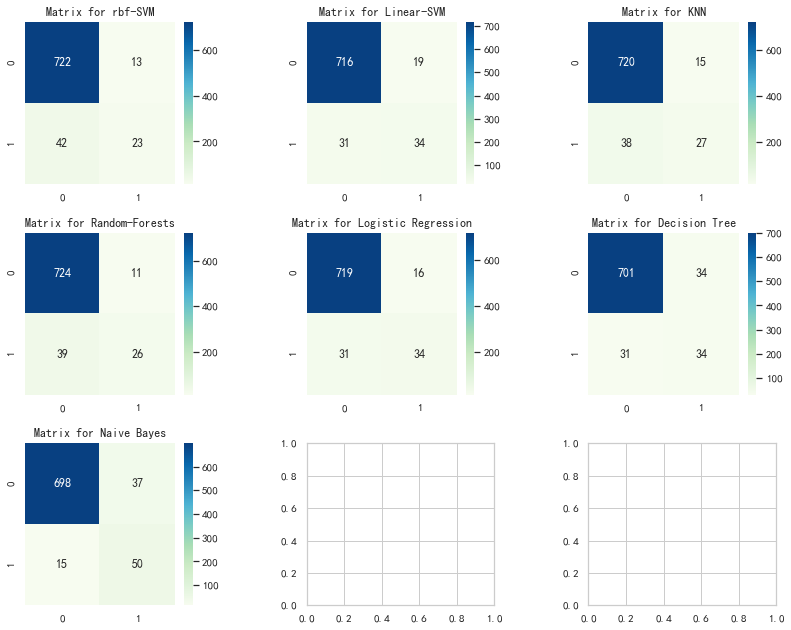
对3.1中建立的所有模型进行10折交叉验证，得到各模型的平均准确率和标准差如下：

|  |  |  |
| --- | --- | --- |
| 模型 | 平均准确率 | 标准差 |
| Linear Svm | 0.93750 | 0.053033 |
| Radial Svm | 0.92875 | 0.063258 |
| Logistic Regression | 0.94375 | 0.048170 |
| KNN | 0.92625 | 0.060583 |
| Decision Tree | 0.90375 | 0.076659 |
| Naive Bayes | 0.92875 | 0.039151 |
| Random Forest | 0.93125 | 0.055410 |

画出箱线图如下：



各模型对应的混淆矩阵图如下：



总体而言，在所有模型中，逻辑回归的表现最好，准确率集中在90%以上。决策树的表现最差，准确率波动相对较大，最低时只有85%。朴素贝叶斯分类最为稳定，准确率集中在92%左右。

### 集成算法

#### Voting Classifier

将3.1中的模型集中起来，同时进行预测并投票，少数服从多数即为Voting Classifier.

10折交叉验证下Voting Classifier的准确率达到了0.94375，比3.1中每个单独模型的交叉验证准确率都高，故投票法集成是有效的。

#### Bagging

给定一个大小为n的训练集D，Bagging算法从中均匀、有放回地（即使用自助抽样法）选出m个大小为n'的子集Di，作为新的训练集。在这m个训练集上使用分类、回归等算法，则可得到m个模型，再通过取平均值、取多数票等方法，即可得到Bagging的结果 。

对KNN模型进行Bagging,10折交叉验证的准确率从0.92625提高到了0.92875，虽然不明显，但是有了一定改善。

#### Boosting

Boosting是一种框架算法,主要是通过对样本集的操作获得样本子集,然后用弱分类算法在样本子集上训练生成一系列的基分类器。他可以用来提高其他弱分类算法的识别率,也就是将其他的弱分类算法作为基分类算法放于Boosting 框架中,通过Boosting框架对训练样本集的操作,得到不同的训练样本子集,用该样本子集去训练生成基分类器;每得到一个样本集就用该基分类算法在该样本集上产生一个基分类器,这样在给定训练轮数 n 后,就可产生 n 个基分类器,然后Boosting框架算法将这 n个基分类器进行加权融合,产生一个最后的结果分类器,在这 n个基分类器中,每个单个的分类器的识别率不一定很高,但他们联合后的结果有很高的识别率,这样便提高了该弱分类算法的识别率。在产生单个的基分类器时可用相同的分类算法,也可用不同的分类算法,这些算法一般是不稳定的弱分类算法,如神经网络(BP) ,决策树(C4.5)等。

##### AdaBoost

Adaboost是一种迭代算法，其核心思想是针对同一个训练集训练不同的分类器(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器（强分类器）。

迭代200次，设定学习率为0.1，最终Adaboost的10折交叉验证准确率为0.93875，超过了除逻辑回归以外所有模型的准确率。

##### Gradient Boosting

AdaBoost使用的是指数损失函数，这个损失函数的缺点是对于异常点非常敏感，，因而通常在噪音比较多的数据集上表现不佳。Gradient Boosting（梯度提升）在这方面进行了改进，使得可以使用任何损失函数 (只要损失函数是连续可导的)，使模型抗噪音能力更强。

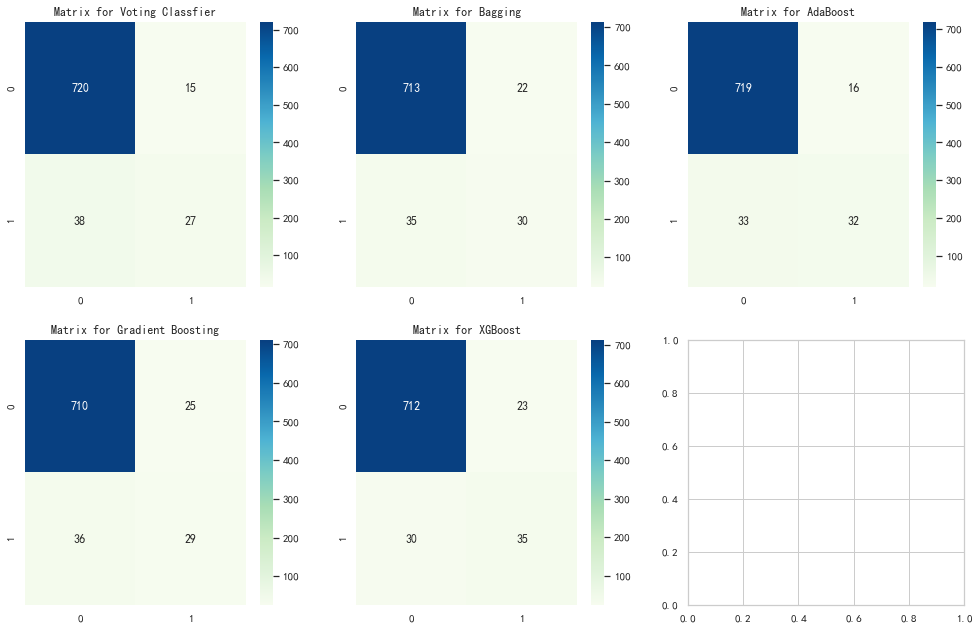
设定学习率为0.1，Gradient Boosting的10折交叉验证精度为0.92375。

##### XGBoost

XGBoost继续在Gradient Boosting的基础上对损失函数进行改进，一是在原损失函数的基础上添加了正则化项产生了新的目标函数，这类似于对每棵树进行了剪枝并限制了叶结点上的分数来防止过拟合。二是对目标函数进行二阶泰勒展开，以类似牛顿法的方式来进行优化。

设定学习率为0.1，Gradient Boosting的10折交叉验证精度为0.93374。

#### 精度对比



对比各集成算法和原模型，发现相较于原模型，集成算法的准确率都有了一定提高，其中Voting Classfier和AdaBoost的准确率相对较高，分别达到了94.375%和93.875%。

### 特征重要性

特征重要性评分是一种为输入特征评分的手段，其依据是输入特征在预测目标变量过程中的有用程度。



如上图所示，计算出各集成模型中的特征重要性并可视化，可以看到SP攻击,SP Def和speed是最重要的三个指标,其次为attack和defense。Generation和Type1在各模型中的重要性评分都很低，其中generation在adaboost中完全不重要，即被剔除了。

因此，可以得出结论，特殊攻击、特殊防御和速度是识别传奇宝可梦和普通宝可梦的主要影响因素；世代和主要属性则对宝可梦的分类帮助不大，说明传奇宝可梦和普通宝可梦的主要属性和各世代的分布上差异不大。

## 结论

本文使用了Kaggle上的公开数据集口袋妖怪数据，综合这门课程所学知识，对宝可梦的不同分类（普通宝可梦和传说宝可梦）影响因素进行探究，并得出相关结论。本文首先介绍了宝可梦的相关背景，其次对数据进行了一系列统计描述，展现变量的相关性和分布特点，以及普通宝可梦和传说宝可梦在各个特征上的异同；并对原始数据进行清洗；之后基于数据的特点，建立逻辑回归、SVM、决策树、随机森林、K近邻算法和朴素贝叶斯分类器等模型进行分类，接着采用集成算法对模型做出改进；最终找到影响宝可梦类别的关键因素。研究结果表明，特殊攻击、特殊防御和速度是识别普通宝可梦最关键的三个因素。