预测森林覆盖类型

预测森林覆盖类型

摘要 ABSTRACT

20 世纪 90 年代,世界森林资源遭到严重破坏 ,森林资源大幅度减少，造成森林资源减少的主要原因是自然灾害,毁林开荒以及对林产品需求的增加。最近 20 年来，人类社会对森林的价值的认识有了很大的变化。在欧美等发达国家，人们对于森林的生态、社会和公益价值的需求正在不断增长。对于不同地区的森林类型的保护政策也大相径庭，因此，根据当地现有的生态条件来预测当地的森林类型进行管治保护具有重要的意义。本文旨在利用不同的数据降维方法和分类方法，根据各项特征来预测不同的森林，挖掘与森林类型密切相关的生态指标。本文使用来自 UCI Machine Learning Repository 中提供的 Covertype Data Set 数据集，采取相关数据特征提取和使用 PCA 主成分降维的方式进行特征处理，之后利用 KNN,决策树和随机森林三种分类方法对之前两种降维方式进行了准确性的比较，得到特征提取后的数据分类后的表现优于 PCA 降维。针对特征提取后的数据，又进一步进行了 SVM，逻辑回归与 XGBOOST 三种分类方式，比较不同模型间的分类结果，得到 KNN 的分类准确率最高，为 96.57%，又对 KNN 的相关参数进一步做了调参优化，最终在测试集上取得了 96.86%的准确率，证明本文的方法具有一定准确性。

1. 引言 INTRODUCTION

罗斯福国家公园位于荒烟漫草的北达科他州，曾经有不少的野生动物，但近年来，由于生态条件的恶化，许多地区出现了荒漠化现象。本文研究的是科罗拉多州罗斯福国家森林四个区域的树木观测结果，该研究区域为森林中的四个荒野地区。这些区域代表的是受人为干扰最少的森林，因此现有的森林覆盖类型更多是生态过程的结果，而不是森林管理实践的结果。

本数据集共包含 581012 个数据量，有 54 个特征，比较符合本学期数据挖掘的课程要求。同时，森林覆盖类型的分类研究是森林资源变化监测,森林资源合理开发,森林人工修复的前提条件,因此构建分类准确的森林覆盖类型分类模型具有相当重要的现实意义。

实验过程中，本文首先对数据进行了初步的观察与分析，在特征提取上，为了学习不同特征提取方案的不同结果，采用两种方案，一是对数据集进行偏度分析后将其分割，手动选取其中部分属性特征。二是采用主成分分析法进行降维，手动设置降维至 44 维。

针对特征提取与降维后的数据进行分类，分别使用 KNN 模型，决策树模型与随机森林模型，通过不同分类模型对特征提取后的分类准确度选择一种更好的特征分类方法，比较后发现，手动提取特征后的数据分类表现远优于 PCA 降维后的分类表现。

针对特征提取后的数据，除了上述的三种分类模型外，为了尽可能多的学习不同分类模型的原理与分类表现，掌握面对不同数据与分类要求时不同分类模型的利与弊，本文又使用了 SVM 支持向量机、逻辑回归分类与 XGBOOTS 模型进行分类，在比较六种分类模型的准确度后发现，针对本数据集，KNN 在准确度与运行时间上的表现优于其他模型，更加适合此类数据集的分类工作。

2. 数据 DATA

2.1 数据概述

本文使用的数据来自 UCI Machine Learning Repository 中的 Covertype Data Set 数据集。

数据源:

http://archive.ics.uci.edu/ml/datasets/Covertype

covtype.data 为网站提供的原始数据，本文通过 python 数据处理为每列增加了相应的列名，导出 covtype.csv 作为后续处理的数据集。

该数据集包含科罗拉多州罗斯福国家森林四个区域的树木观测结果。 相关信息见下表：

表格 1 数据集有关信息

数据量	581012	属性数量	54
相关任务	分类	有无缺失值	无
数据集特征	多变量	属性特征	整数、分类

属性特征包括海拔，方位角纵横比，坡度，土壤类型等，共 54 维，其中 Wilderness Area 与 Soil Type 属于 one-hot 编码。

最后一列 cover_type 代表该地区的森林类型对应的数字，在这个数据集下统计不同类型的森林数量如下所示，其中森林类型共计 7 种分别是：

1 - Spruce/Fir
2 - Lodgepole Pine
3 - Ponderosa Pine
4 - Cottonwood/Willow
5 - Aspen
6 - Douglas-fir
7 - Krummholz

特征字段的数据都是数值型数据，要预测的森林类型也是数值型数据，因此都可以不用进行类型转换，可以直接使用。并且数据集完整，没有缺失的数据，因此只对数据集进行数据处理即可。

2.2 数据分析及可视化

首先在 jupyter 对数据集进行统计分析以及可视化操作。如下：

2.2.1 特征数据分布统计

调用 describe()函数检查属性分布信息，下图仅展示部分信息。

[13]:

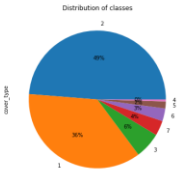
	elevation	aspect	slope	horizontal_distance_to_hydrology	vertical_distance_to_hydrology	horizontal_distance_t
count	581012.000000	581012.000000	581012.000000	581012.000000	581012.000000	58
mean	2959.365301	155.656807	14.103704	269.428217	46.418855	
std	279.984734	111.913721	7.488242	212.549356	58.295232	
min	1859.000000	0.000000	0.000000	0.000000	-173.000000	
25%	2809.000000	58.000000	9.000000	108.000000	7.000000	
50%	2996.000000	127.000000	13.000000	218.000000	30.000000	
75%	3163.000000	260.000000	18.000000	384.000000	69.000000	
max	3858.000000	360.000000	66.000000	1397.000000	601.000000	

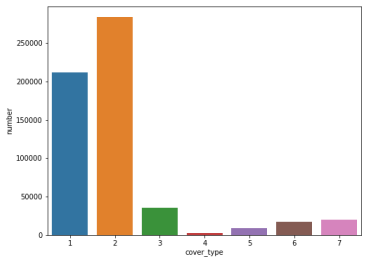
8 rows × 55 columns

以上信息说明各属性值的分布以及尺度比例是不统一的，需要后续需要缩放处理。

2.2.2 森林分布统计

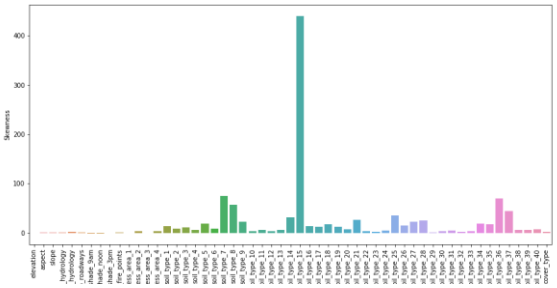
统计发现，该区域类型 1，2 的森林占主导地位；类型 4 的森林数目最少。





2.2.3 数据偏度分析

偏度（skewness）也称为偏态、偏态系数，是统计数据分布偏斜方向和程度的度量，是统计数据分布非对称程度的数字特征。调用 skew()函数对各属性进行偏度分析结果如下：



由偏度统计得一些属性特征具有严重偏斜情况，因此需要在以后的阶段进行更正或转换。

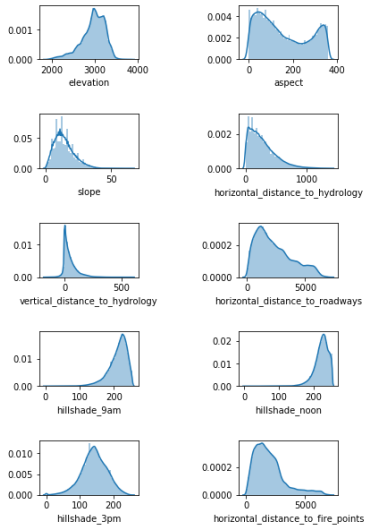
2.2.4 数据集分割

由于 one-hot 类型属性的存在，为了具体了解数据集的特性，基于以下规则划分数据集并进行相应分析。

- cont_data – 仅包含不是 one-hot 编码的数据
- binary_data – 仅包含是热编码的数据
- wilderness_data – 仅包含 Wilderness Areas 数据
- Soil_data – 仅包含 Soil Types

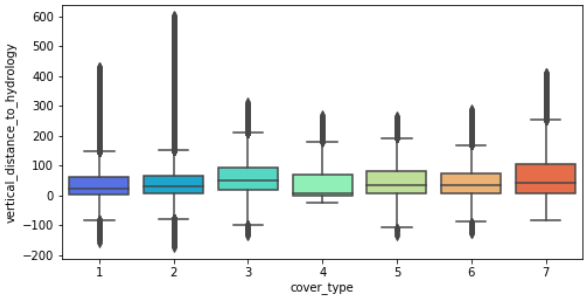
➢ cont_data 数据集分析

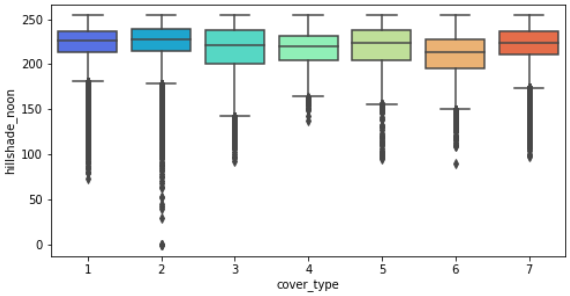
- 直方图分析：只从数据集中获取一个变量，并显示每次出现的频率。



以上直方图表明属性特征的数据值分布存在偏差。

- 箱型图分析：方框图显示了数据的分布和更详细的信息。它更清楚地显示了异常值：最大值、最小值、四分位数（Q1）、第三四分位数（Q3）、四分位数范围（IQR）和中值



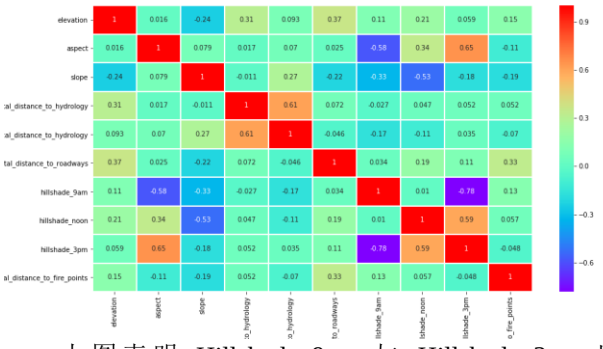


而箱型图表明存在一些特征属性对于不同森林类型不敏感。如 hillshade_noon,vertical_distance_to_hydrology 等。

- 相关性系数的热力图分析

在统计学中，皮尔逊积矩相关系数（用于度量两个变量 X 和 Y 之间的相关程度（线性相关），其值介于-1 与 1 之间。

```
corr(hillshade_9am and hillshade_3pm )= -0.78
corr(aspect and hillshade_3pm )= 0.65
corr(horizontal_distance_to_hydrology and vertical_distance_to_hydrology )= 0.61
corr(hillshade_noon and hillshade_3pm )= 0.59
corr(aspect and hillshade_9am )= -0.58
corr(slope and hillshade_noon )= -0.53
```

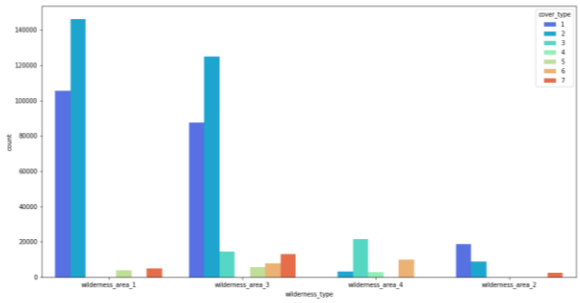


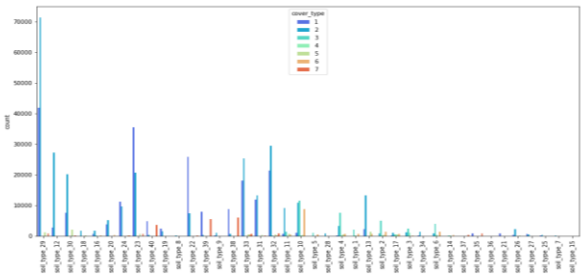
上图表明 Hillshade_9am 与 Hillshade_3pm 以及 Aspect 以及 Hillshade_3pm 的相关性系数较高，分别为-0.78 以及 0.65。

➢ binary_data 数据集分析

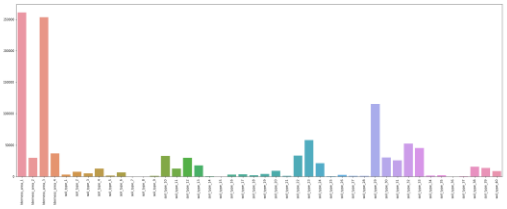
● 直方图分析

通过直方图分析荒野类型以及土壤类型分别与森林类型的关系：





本文对二进制属性集数据进行统计，如下图所示：



分析得土壤类型出现次数表现出极大的不平衡性，进一步筛选出小于 1000 的类型如下：

属性	次数
soil_type_7	105
soil_type_8	179
soil_type_14	599
soil_type_15	3
soil_type_21	838
soil_type_25	474
soil_type_28	946
soil_type_36	119
soil_type_37	298

● PairGrid

PairGrid 将数据集上的每个变量映射到网格的行和列上，便于实现数据信息的快速提取。

2.3 数据处理

2.3.1 特征筛选

通过上述的数据分析，最终确定删除属性特征为'hillshade_3pm','soil_type_7','soil_type_8','soil_type_14','soil_type_15','soil_type_21','soil_type_25','soil_type_28','soil_type_36','soil_type_37'的项。

2.3.2 数据集与训练集的划分

如果模型只是在训练集上表现得较好，而对测试集的效果差强人意，其实是一种过拟合的表现。因此，需要把数据划分训练集和测试集，分别进行分类预测正确性的评估来证明本文的模型具有较强的鲁棒性。

训练集和测试集的划分方式如下：

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3, random_state=123)

2.4 数据降维

为了同数据分析后进行特征筛选的结果进行对比，使用 PCA 主成分分析对数据进行降维。

PCA(Principal Component Analysis)，即主成分分析方法，主要思想是将 n 维特征映射到 k 维上，这 k 维是全新的正交特征也被称为主成分，是在原有 n 维特征的基础上重新构造出来的 k 维特征。PCA 的工作就是从原始的空间中顺序地找一组相互正交的坐标轴，忽略包含方差几乎为 0 的特征维度，实现对数据特征的降维处理。

通过 PCA(n_components = 44)手动设置将数据降维至 44 维。用于比较特征筛选后的分类结果。

3. 方法 methods

3.1 K-近邻算法

K-近邻算法是一种用于分类和回归的非参数统计方法。训练样本是多维特征空间向量，其中每个训练样本带有一个类别标签。算法的训练阶段只包含存储的特征向量和训练样本的标签。在分类阶段，k 是一个用户定义的常数。一个没有类别标签的向量（查询或测试点）将被归类为最接近该点的 k 个样本点中最频繁使用的一类。

确定 KNN 的参数如下：

```
KNeighborsClassifier(algorithm='auto',leaf_size=30,
metric='minkowski',metric_params=None,n_jobs=None,
n_neighbors=3, p=2,weights='uniform')
```

为了进一步分析特征筛选的效果，对耗时以及准确率进行分析，如下

	未进行特征提取	进行特征提取	PCA 降维
耗时(s)	19.6	17.4	26.8
Accuracy	0.966242	0.965686	0.966242

进行特征提取与未进行特征提取在准确率上的差距仅为 0.05%，进行特征提取的数据减少了 10 维的特征，耗时上缩减了 12.6%。可以认为特征提取在保证较高正确率的同时，极大地提高了算法效率

3.2 决策树

机器学习中，决策树是一个预测模型；他代表的是对象属性与对象值之间的一种映射关系。树中每个节点表示某个对象，而每个分叉路径则代表的某个可能的属性值，而每个叶结点则对应从根节点到该叶节点所经历的路径所表示的对象的值。决策树仅有单一输出，若欲有复数输出，可以建立独立的决策树以处理不同输出。数据挖掘中决策树是一种经常要用到的技术，可以用于分析数据，同样也可以用来作预测。

3.2.1 结果分析

确定决策树的参数如下：

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

为了进一步分析特征筛选的效果，对耗时以及准确率进行分析，如下

	未进行特征提取	进行特征提取	PCA 降维
耗时(s)	7.07	6.67	69
Accuracy	0.9345912887828163	0.9352166330089958	0.9190093170552598

进行特征提取的数据集的训练模型结果正确率优于未进行特征提取的训练集模型，且耗时更短。可以认为特征提取在保证较高正确率的同时，极大地提高了算法效率

3.3 随机森林

原理：建立多个决策树并将他们融合起来得到一个更加准确和稳定的模型，是 bagging 思想和随机选择特征的结合。随机森林构造了多个决策树，当需要对某个样本进行预测时，统计森林中的每棵树对该样本的预测结果，然后通过投票法从这些预测结果中选出最后的结果。

随机体现在两个方面，一个是随机取特征，另一个是随机取样本，让森林中的每棵树既有相似性又有差异性。

森林中每棵树按照如下方式生长：

1.如果过训练样本中有 N 个样本，那么从这 N 个样本中有放回的抽样 N 次，将得到的样本用于建树

2.设 M 为输入样本的特征数，对于每个节点分裂时，先从这 M 个特征中选择 m(m<<M)个特征，然后再在这 m 个特征中选择最佳的分裂点进行分裂

3.每棵树都尽可能的生长，没有剪枝

m 的值越大，上述 1 中的相关性越高，2 中的分类能力也越强，所以 m 在 RF 中是一个非常重要的参数。

随机森林的预测错误率取决于以下两点：

1.森林中任意两棵树之间的相关性，相关性越高，错误率越大

2.每棵树的分类能力，单棵树的分类能力越强，那么整个森林的分类能力也越强

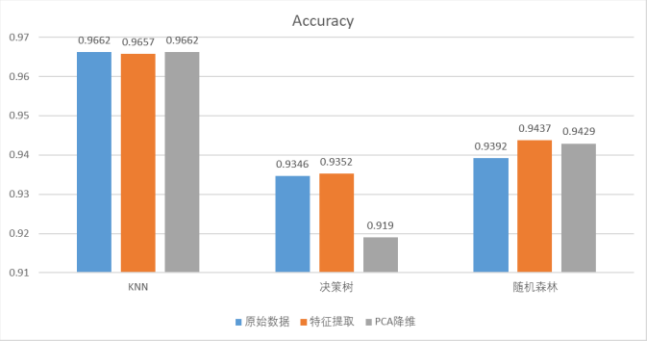
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

	未进行特征提取	进行特征提取	PCA 降维
耗时(s)	12.1	10.7	5.52
Accuracy	0.9392	0.9437	0.9429

3.4 特征提取与 PCA 降维的比较

本次实验数据经过特征提取与 PCA 降维后，数据维度由 54 维降至 44 维，针对 PCA 降维与特征提取后的数据，本文分别使用 KNN 算法，决策树算法，随机森林算法进行分类汇总分类结果如下：

Accuracy	原始数据	特征提取	PCA 降维
KNN	0.9662	0.9657	0.9662
决策树	0.9346	0.9352	0.9190
随机森林	0.9392	0.9437	0.9429



从分类结果的精确度与运行时间可以看出，特征提取后的分类表现优于 PCA 降维，本文决定再选用几种分类方法对特征处理后的数据进行分类，比较不同分类方法在此数据集中的表现。

3.5 SVM 分类

支持向量机（Support Vector Machine, SVM）是一类按监督学习方式对数据进行二元分类的广义线性分类器。

使用支持向量机处理未降维的数据，设置参数为 SVC(random_state = 0)，处理数据的过程中发现，由于此次数据量偏大，使用 SVM 分类方法的运行时间过长，失去训练意义，因此淘汰本分类模型。

3.6 逻辑回归

logistic 回归是一种广义线性回归（generalized linear model），因变量可以是二分类的，也可以是多分类的，但是二分类的更为常用。

此次分类并非二分类，可以预见到，逻辑回归处理得到的分类精度并不高，但是这种经典分类算法应该被学习使用。

使用逻辑回归方法处理特征处理后的结果，得到逻辑回归的处理结果。

耗时(s)	19.43s
Accuracy	0.6140306590783918

3.7 XGBOOST

极端梯度提升(XGBoost)[5]。它是一个经过设计和优化的库用于增强的树算法。XGBoost 是由华盛顿大学的陈天奇发明的集成学习算法，因其在回归问题上的表现优异，在 Kaggle、阿里天池等数据挖掘比赛中被广泛使用。

该算法的主要思想就是不断地添加树，不断地进行特征分裂来生长一棵树。每次添加一个树，其实就是学习一个新函数，去拟合上次预测的残差。当训练完成得到k棵树，要预测一个样本的分数，就是根据这个样本的特征，在每棵树中会找到对应的一个叶子节点，每个叶子节点对应一个分数，最后只需要将每棵树对应的分数求和就是该样本的预测值。

XGBoost 要优化的目标函数为：

$$Obj(t) = \sum_{i=1}^{n} L\left(y_i, \hat{y}^{t-1} + f_t(x_i)\right) + \Omega(f_t) + c$$

其中，L为损失函数，正则项$\Omega(f_t)$为：

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{i=1}^{T} w_j^2$$

其中，T表示叶子节点数，w_j表示第j个叶子节点的权重。

使用 XGBOOT 方法处理特征处理后的结果，得到处理结果。

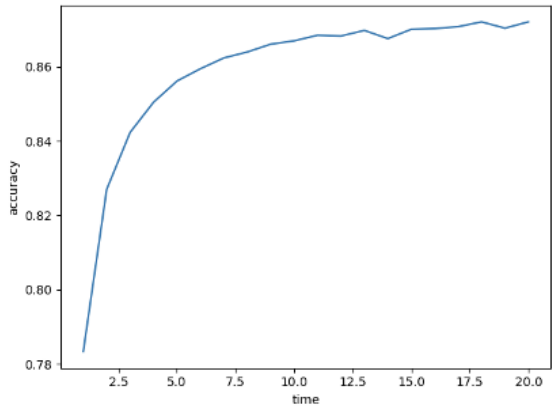
耗时(s)	8.73min
Accuracy	0.8725

3.8 神经网络

使用 add()函数添加层，首先添加输入层，input_shape 用于指定输入数据的维度，激活函数使用 relu，接着使用 Dropout()随机舍去 20%的节点，舍去节点的目的是为了防止 overfitting，最后添加输出层，节点为 8 个，使用 softmax 函数处理输出，设置损失函数与优化方法为 adam，构建的网络结构如下所示：

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 200)               11000
_________________________________________________________________
dense_1 (Dense)              (None, 150)               30150
_________________________________________________________________
dropout (Dropout)            (None, 150)               0
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 1208
=================================================================
Total params: 42,358
Trainable params: 42,358
Non-trainable params: 0
_________________________________________________________________
```

训练过程精度提升示意图：



	进行特征提取	未进行特征提取
耗时(s)	641	814
Accuracy	0.8149	0.8895

由结果可以看出，神经网络虽然训练成本高，但是结果并没有 KNN、决策树等准确度高。通过查阅文献与资料，可以知道，神经网络更加适用于维度高、训练样本多的情况，本实验所用样本相对较少，同时由于本实验所用神经网络只有两个隐藏层，并且是全连接的，需要更多后续的调参以及模型的更改，才可能达到较高的准确度。

3.8 KNN 调参过程

对于基础的 KNN 模型，为了进一步提高准确率，使用网格搜索法对主要参数进行调参，得到最终的模型。进行调参的主要参数和参数取值情况如下表所示：

参数名称	调参选择
n_neighbors	[1,7]
weights	uniform、distance
p	[1,5]
algorithm	auto
metric_params	none
n_jobs	1

参数解释：

n_neighbors：默认为 5，就是 kNN 的 k 的值，选取最近的 k 个点。

weights：默认是 uniform，参数可以是 uniform、distance，也可以是用户自己定义的函数。uniform 是均等的权重，就说所有的邻近点的权重都是相等的。distance 是不均等的权重，距离近的点比距离远的点的影响大。用户自定义的函数，接收距离的数组，返回一组维数相同的权重。

p：距离度量公式

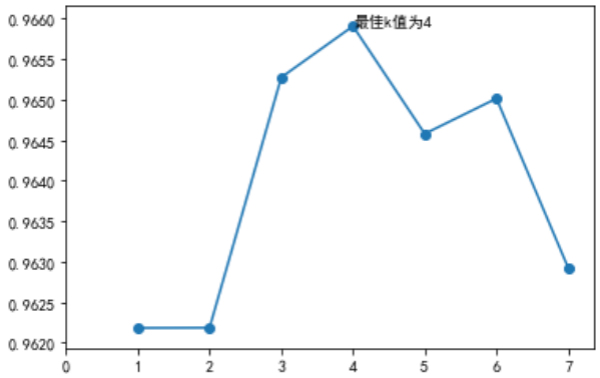
algorithm：快速 k 近邻搜索算法，默认参数为 auto，可以理解为算法自己决定合适的搜索算法。

metric_params：距离公式的其他关键参数，这个可以不管，使用默认的 None 即可。

n_jobs：并行处理设置。默认为 1，临近点搜索并行工作数。如果为-1，那么 CPU 的所有 cores 都用于并行工作。

由于需要调整的参数较多，如果直接进行网格搜索可能需要非常长的时间。因此，采取如下的贪心策略进行调参：按上表的参数顺序进行逐个调整。每调整完一个参数后就将该参数设为当前得到的最佳参数值，以此为基础进行下一个参数的调整。

(1)n_neighbors（k 值）调参



(2)weights 调参

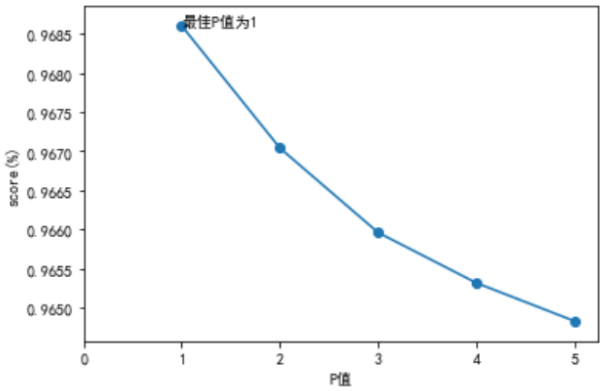
此处用图无法展示，因此对 uniform 和 distance 两种方法下的最高得分进行了一个比较。

```
k=4,得分为:0.9621,最好的方法uniform
k=4,得分为:0.9670,最好的方法distance
最好的k为:4,最好的得分为:0.9670,最好的方法distance
```

不难看出，使用 distance 的方法在此处更为合适。

(3)p 值调参

本文在 k=4，weights=distance 基础上对 p 进行调参确定。




```
最好的k为:4,最好的得分为:0.9686,最好的p为1
```

最终，调参得到的最佳参数如下表所示：

参数名称	调参选择
n_neighbors	4

weights	distance
p	1
algorithm	auto
metric_params	none
n_jobs	none

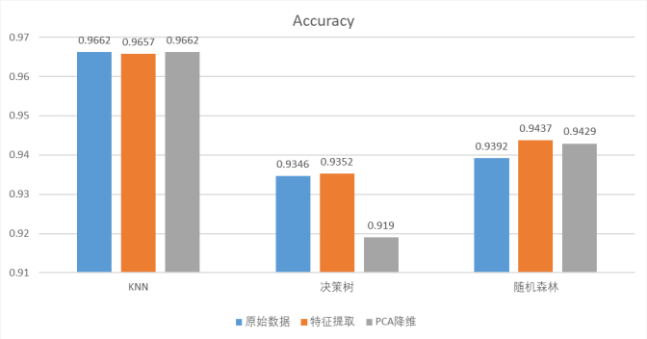
在此基础上，最好的得分结果为 96.86%。相比最开始 n_neighbors=5，weights=uniform，p=2 下最高的准确率 96.57%也产生了进一步的效果提升，证明本文的工作是有意义的。

4. 结果 RESULTS

本次实验通过对现有森林植被类型进行分析与特征提取，预测未来森林中植被的覆盖类型。

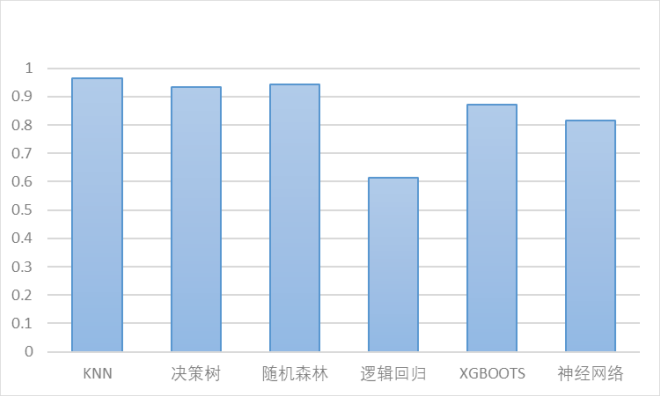
在对数据集中的数据进行初步观察与分析后，共采用两种方式对数据降维。一是根据数据偏度与相关性进行分析后手动删除部分特征，同时采用 PCA 将原数据集降至相同的维度，比较降维结果。

降维后的数据使用 KNN 算法、决策树算法、随机森林算法进行分类，分类准确度如下图：



在分类后可知，针对该数据集，特征提取的表现优于 PCA 降维，其中，KNN 算法分类的准确性最高，达到 96%。

为了比较不同分类方法之间的差异，又使用了四种分类方法，分别是 SVM、逻辑回归、XGBOOTS 与神经网络对特征提取后的数据进行分类，其中 SVM 因为耗时太长被淘汰，其他分类方法的分类结果如下：



在六种分类方法中，KNN 的分类准确度最高，因此本文对 KNN 进行进一步的调参，希望能提升准确率。经过调参后的 KNN 分类准确率由 96.57%提高至 96.86%，准确度进一步提升。

5. 结论 CONCLUSION

通过本次研究，本文对数据挖掘和分析的过程有了更为深刻的理解，对不同的降维算法和分类算法有了更深刻的了解。在实验过程中，最深刻的体会是不同分类方法选取决定了最终预测结果的好坏，而模型调参只是逼近分类预测正确率的上限。因此，对实验方法进行恰当的选择是非常必要的。

此外，对于训练集和测试集要做相同的特征提取处理。起初由于测试集中特征的提取与训练集中不一致造成模型在测试集上的效果比较差。在今后做类似的数据挖掘时，要注意代码的版本管理。起初做该实验时代码的存放比较随意，导致后续多个版本的代码放在一起造成了一定的混乱。最好按照不同的分类进行分别管理，这样可以在后续实验中能够及时与之前的代码结果进行对比。最后，虽然实验过程中遇到了各种各样的问题，但是通过小组成员之间的合作也最终解决，可以说是受益匪浅。

参考文献 REFERENCES

[1] GENG Li-juan, LI Xing-yi. 用于大数据分类的 KNN 算法研究[J]. 计算机应用研究, 2014, 31(005):1342-1344,1373.

[2] 周志华. 机器学习. 清华大学出版社, 2016.

[3] 韩家炜, 坎伯. 数据挖掘: 概念与技术. Vol. 2. No. 007. 机械工业出版社, 2012.

[4] 曹正凤. 随机森林算法优化研究[D]. 首都经济贸易大学.

[5] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.

[6] Xgboost developers. "XGBoost Documentation." https://xgboost.readthedocs.io/en/latest/.