数据挖掘可视化系统-设计文档 V1.0

作　　者：　许继元 程络 黄倬熙 张平路

刘浩斌 邓志聪

学　　历：　本科

指导老师 ：　谢光强

目录

1. 设计分析

随着 5g 网络的兴建，国家高度重视人工智能的发展并专门为人工智能产业提出发展规划，人工智能领域将成为时代潮流的风口浪尖。对人工智能人才的缺口也会急速上升。数据挖掘作为人工智能的分支之一，也吸引了诸多学子。可是，抽象的算法和复杂的编程却让人却让人望而却步。

无论是想利用数据挖掘解决问题的编程小白，还是想要学习数据挖掘的菜鸟，复杂的 python 语言和抽象的机器学习算法都是横亘在他们面前的巨石。

对于本身不会编程但想要利用数据挖掘解决问题的人，无论是繁琐的编译器配置，以及漫长的 python 语言学习周期，都阻止他解决迫在眉睫的数据挖掘问题。

对于略有编程基础想要学习数据挖掘算法的人来说，即使有网上的课程，但缺乏可视化展示，依然会对数据挖掘学习产生阻碍。

目前市面还没有相应的工具，来解决此类的痛点。

本项目实现了数据挖掘的可视化功能，（支持用户自主上传数据集），提供多种数据集以及算法，以及简单方便的 UI 交互功能，让数据挖掘变得简单。

1. 支持多种数据集

本项目将提供多种数据集进行算法分析，如。。。（待补充），并且用户可以自由选择不同特征进行查看。

2. 多样化原始数据可视化方式

在原始数据中，多维数据十分常见，我们采用平行坐标图，雷达图，散点图等多种表现形式多元化，创新性的将原始数据可视化，使得用户能够直观的感受多维数据。

3. 结合多种数据挖掘算法

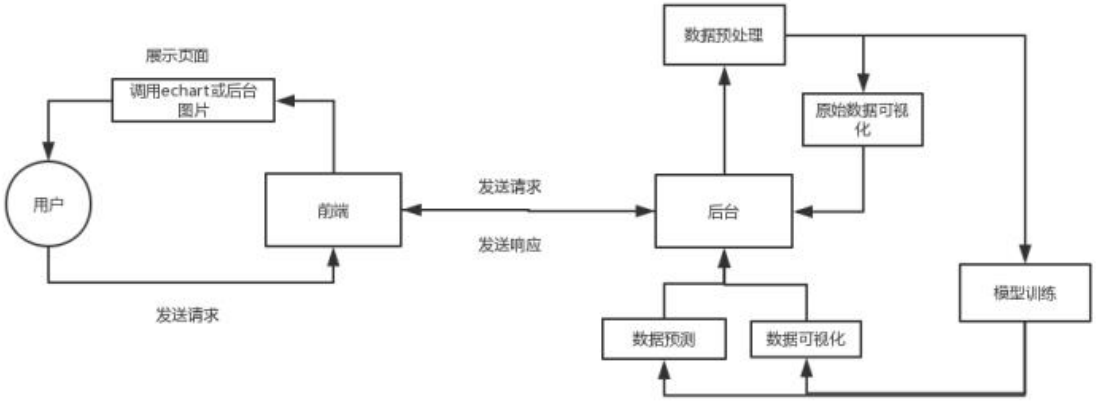
本项目将提供 SVM，多元线性回归，朴素贝叶斯，决策树等多种算法供用户进行选择，以不同方式挖掘出数据背后的规律，理解算法原理。

4. 个性化数据挖掘可视化

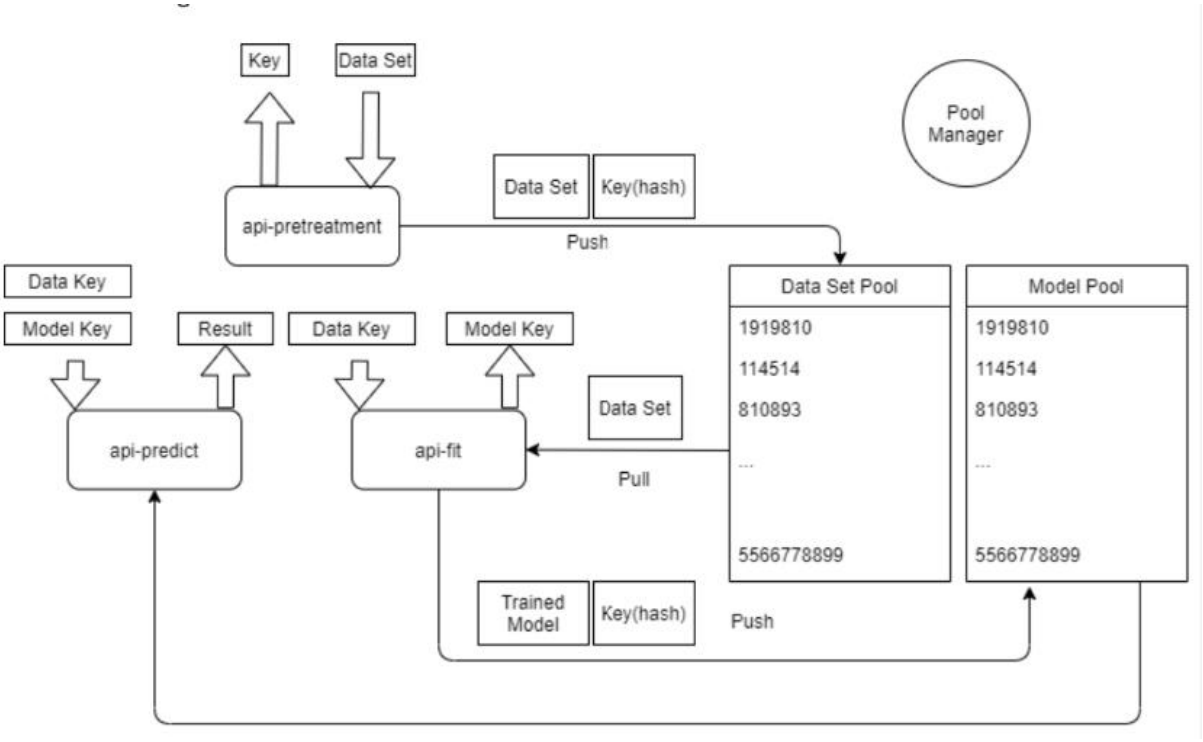
针对不同算法，我们将给出个性化的数据可视化方式，如决策树算法我们将以树状图的形式进行呈现，朴素贝叶斯，我们将使用词云图来达到可视化的目的

2. 系统结构

系统主要由前端和后台组成。前端主要负责和用户的交互，以及数据可视化的呈现。后台部分主要负责数据预处理，模型训练，预测以及可视化。



后台的调度系统的工作流程如下图所示。



3. 系统实现

3.1 实现工具

操作系统：windows10

编程语言：python，html，css，js

开发环境：pycharm

3.2 数据挖掘分类算法

3.2.1 KNN 算法

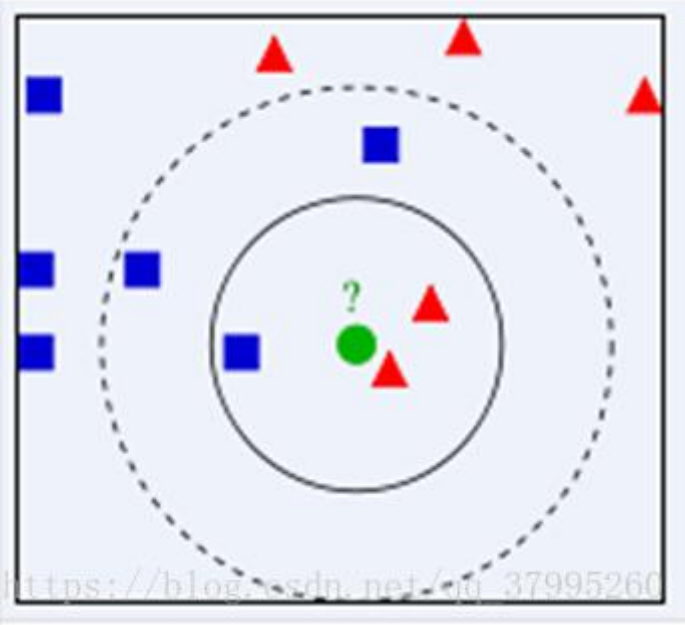
实现目的：

利用 KNN 算法对建立模型并对数据预测实现可视化

实现原理：

kNN 算法的核心思想是如果一个样本在特征空间中的 k 个最相邻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，并具有这个类别上样本的特性。该方法在确定分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。

如下图，绿色圆要被决定赋予哪个类，是红色三角形还是蓝色四方形？如果 K=3，由于红色三角形所占比例为 2/3，绿色圆将被赋予红色三角形那个类，如果 K=5，由于蓝色四方形比例为 3/5，因此绿色圆被赋予蓝色四方形类。



实现过程：

1. 初始化训练集和类别；

2. 计算测试集样本与训练集样本的欧氏距离；

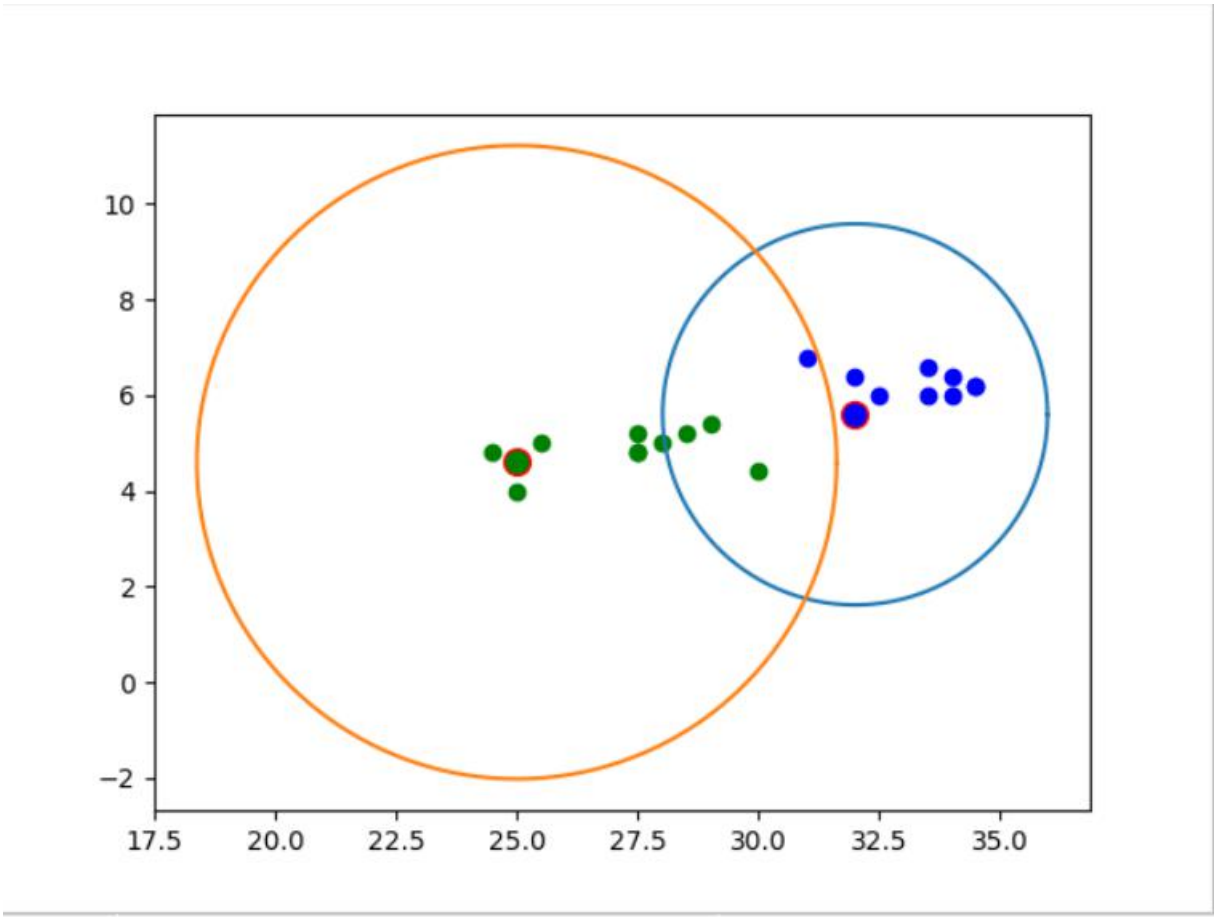
3. 根据欧氏距离大小对训练集样本进行升序排序；

4. 选取欧式距离最小的前 K 个训练样本，统计其在各类别中的频率；

5. 返回频率最大的类别，即测试集样本属于该类别。

实现效果：

使用户能够查看模型评估结果以及可视化结果。



3.2.2 朴素贝叶斯算法

实现目的：

实现对文本（如垃圾邮件）的词库建立，以及新文本预测，并实现算法可视化。

实现原理：

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

P(A)是先验概率，表示每种类别分布的概率；

P(B|A)是条件概率，表示在某种类别前提下，某事发生的概率；该条件概率可通过统计而得出，这里需要引入极大似然估计概念，详见后文。

P(A|B)是后验概率，表示某事发生了，并且它属于某一类别的概率，有了这个后验概率，便可对样本进行分类。后验概率越大，说明某事物属于这个类别的可能性越大，便越有理由把它归到这个类别下。

实现步骤：

1. x={a1,a2,....,am}x =\{a_1,a_2,....,a_m\}$x=\{a_1,a_2,....,a_m\}$为待分类项， 每个 a 为 x 的一个特征属性

2.有类别集合$C= {y_1,y_2,...,y_n} $

3.计算 P(y1|x),P(y2|x),...,P(yn|x)P(y_1|x),P(y_2|x),...,P(y_n|x)

如果 P(yk|x)=max{P(y1|x),P(y2|x),...,P(yn|x)}

实现结果：

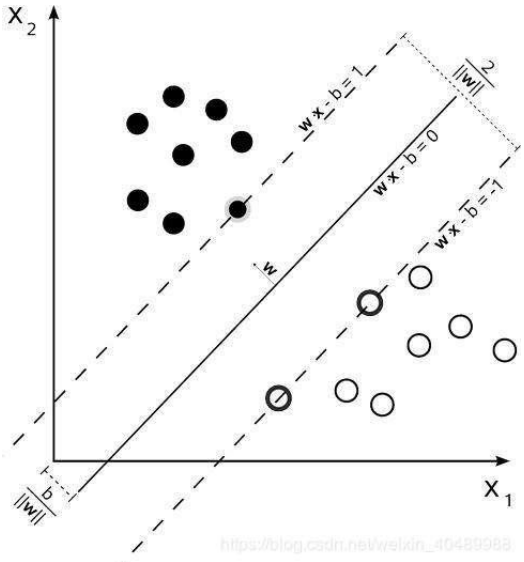
3.2.3 支持向量机算法

实现目的：

寻找到超平面并对数据进行分类，以及实现数据可视化。

实现原理：

超平面和最近的数据点之间的间隔被称为分离边缘，用 P 表示。支持向量机的目标是找到一个特殊的超平面，对于这个超平面分离边缘 P 最大。在这个条件下，决策曲面称为最优超平面。



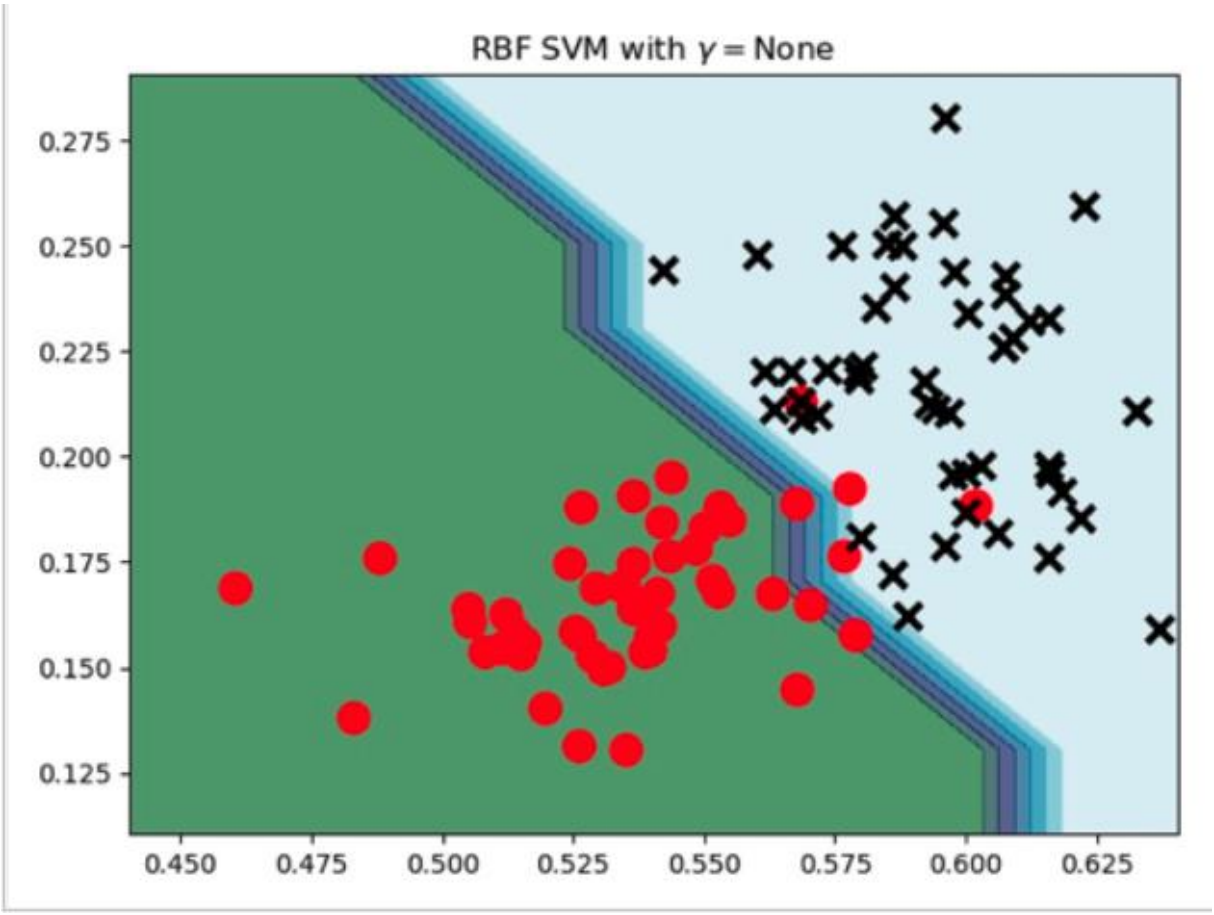
实现步骤：

1. 将寻找最优划分直线问题转化为凸函数优化求解。

2. 引入拉格朗日算子进行求导，将问题再次转换为另一个凸函数求解。

3. 引入松弛变量，通过 SMO 算法，利用多次迭代得到最佳的结果。

实现结果：



3.3 数据挖掘预测算法

3.3.1 多元线性回归

实现目的：

求出多元线性回归的回归方程，对数据进行预测并实现可视化

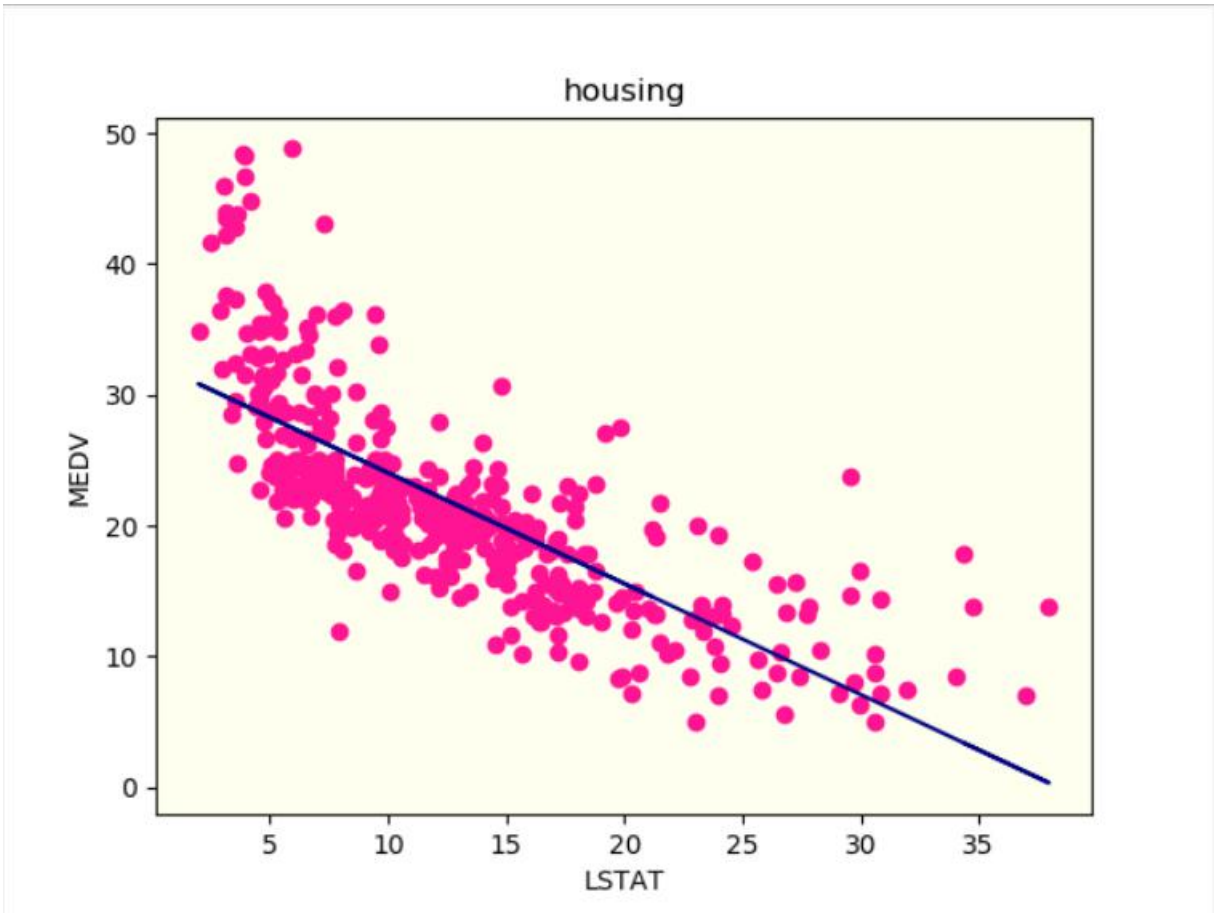
实现原理：

在统计学中，线性回归方程是利用最小二乘函数对一个或多个自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，大于一个自变量的情况叫多元回归。

实现步骤：

1. 将数据矩阵展为增广矩阵
2. 利用最小二乘法求出权重
3. 带入测试集进行检验

实现结果：



3.3.2 逻辑回归

实现目的：

建立逻辑回归的模型，求得回归方程，实现数据的预测以及可视化。

实现原理：

逻辑回归假设数据服从伯努利分布，通过极大化似然函数的方法，运用梯度下降来求解参数，来达到将数据二分类的目的。

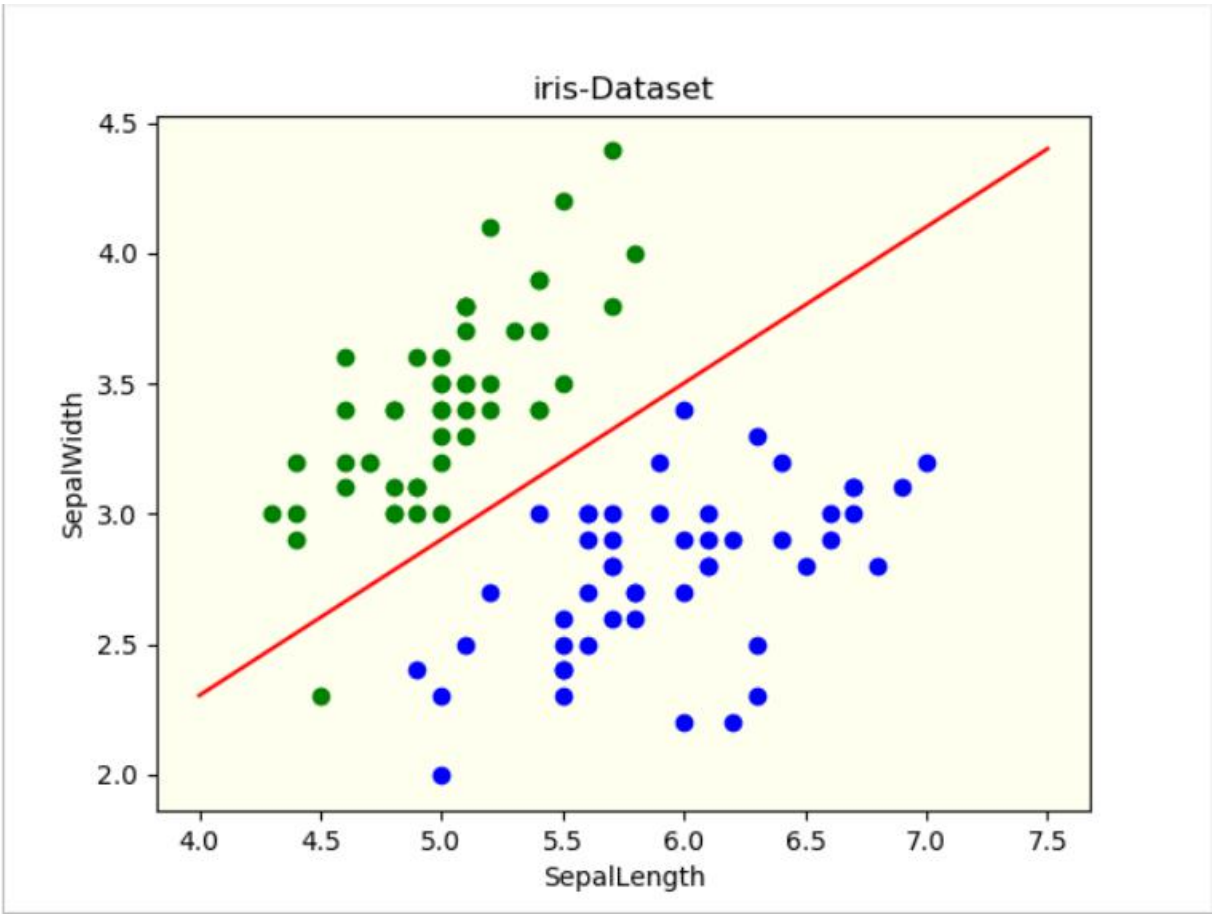
实现过程：

1.寻找一个合适的预测函数,一般是 h 函数(即 hypothesis 函数)。这个函数就是我们要找分类函数

2.构造一个 Cost 函数(即损失函数)，该函数用来表示预测函数（h）与训练数据类别（y）之间的偏差。

3.多次迭代求得最佳的权重，建立最合适的回归模型。

实现结果：



3.3.3 决策树

实现目的：

找到决策树的决策过程，并实现数据的可视化。

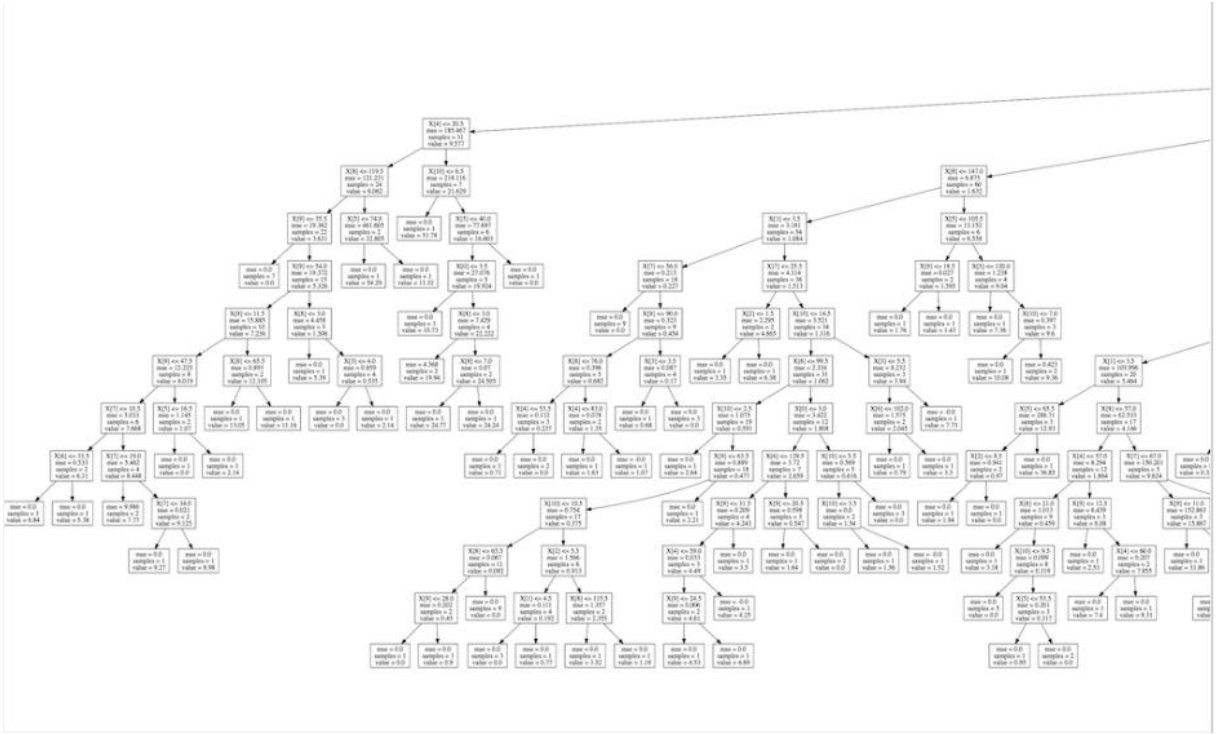
实现原理：

决策树（Decision Tree）是在已知各种情况发生概率的情况下，通过构成决策树来求取净现值的期望值大于 0 的概率，是直观运用概率分析的一种图解法。通俗的讲，决策树就是带有特殊含义的数据结构中的树结构，其每个根结点（非叶子结点）代表数据的特征标签，根据该特征不同的特征值将数据划分成几个子集，每个子集都是这个根结点的子树，然后对每个子树递归划分下去，而决策树的每个叶子结点则是数据的最终类别标签。

实现步骤：

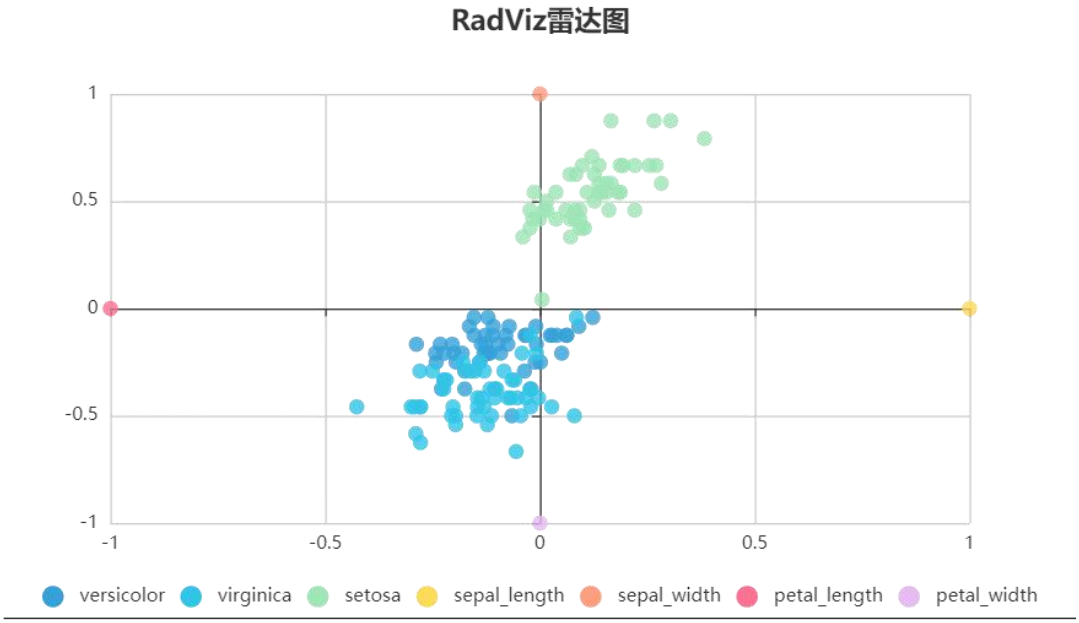
1. 找到基尼系数最小的特征划分
2. 去除已经划分的特征，迭代步骤一。
3. 不断递归，开枝散叶生成决策树。

实现结果：



3.4 多维数据可视化的实现

3.4.1 RadViz 雷达图高维数据显示

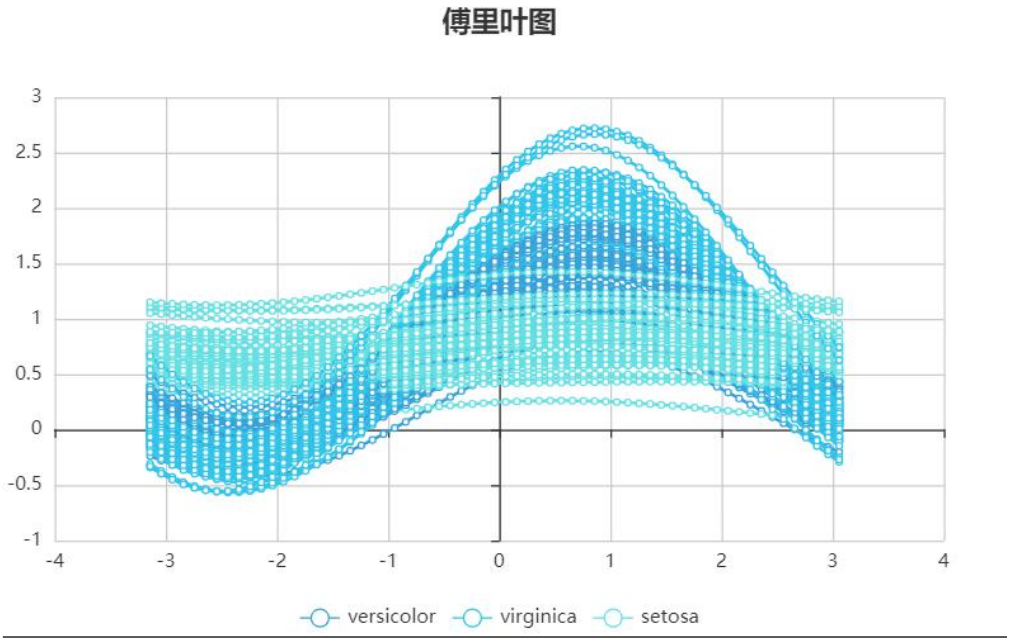


iris 数据集的 RadViz 雷达图-图 3.4.1

基本原理：对所有特征进行归一化之后，通过特征数量为每一个特征在 360 度均分为不

同的轴作为不同特征（维度）的基底向量，将每一个的样本的所有基底向量叠加起来即得到了 RadViz。

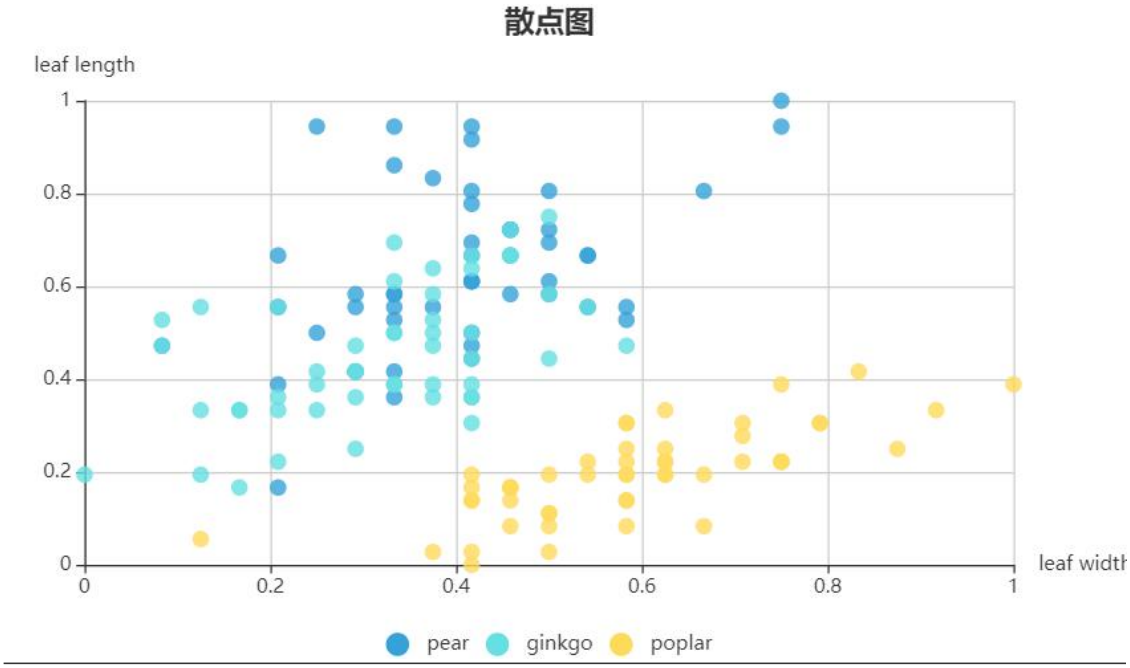
3.4.2 安德烈曲线傅里叶级数图高维数据显示



iris 数据集的安德烈曲线傅里叶级数图-图 3.4.2

生成一个有限项数的傅里叶级数，如何为每一个项的系数指派为一个特征，以此为每一个样本生成一个周期内的傅里叶级数。每一条曲线都代表一个样本，曲线与曲线之间的分离长度象征不同种类之间的差异程度。

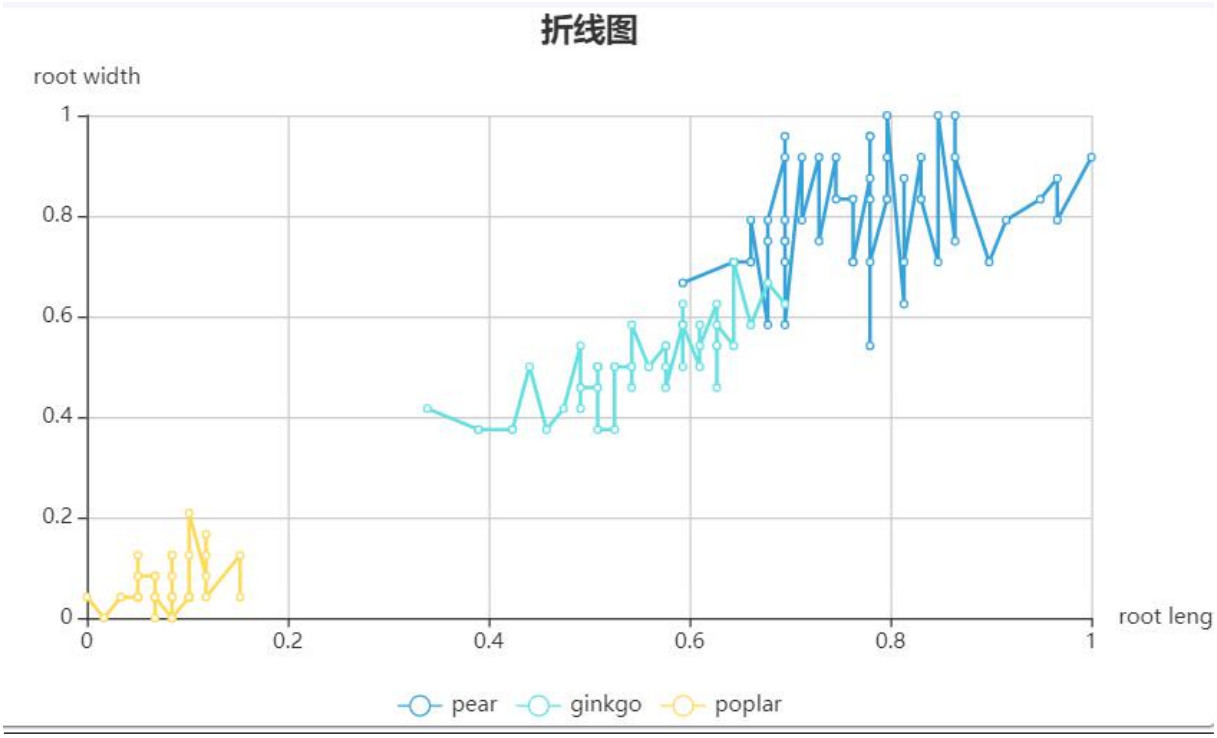
3.4.3 散点图高维数据显示



iris 数据集的散点图-图 3.4.3

每次随机选取数据集中的两个连续特征，作为 x 轴与 y 轴生成散点图，以显示种类分布受到所有不同特征的二维支配，通过低维映射高维的方式展现出联合分布。

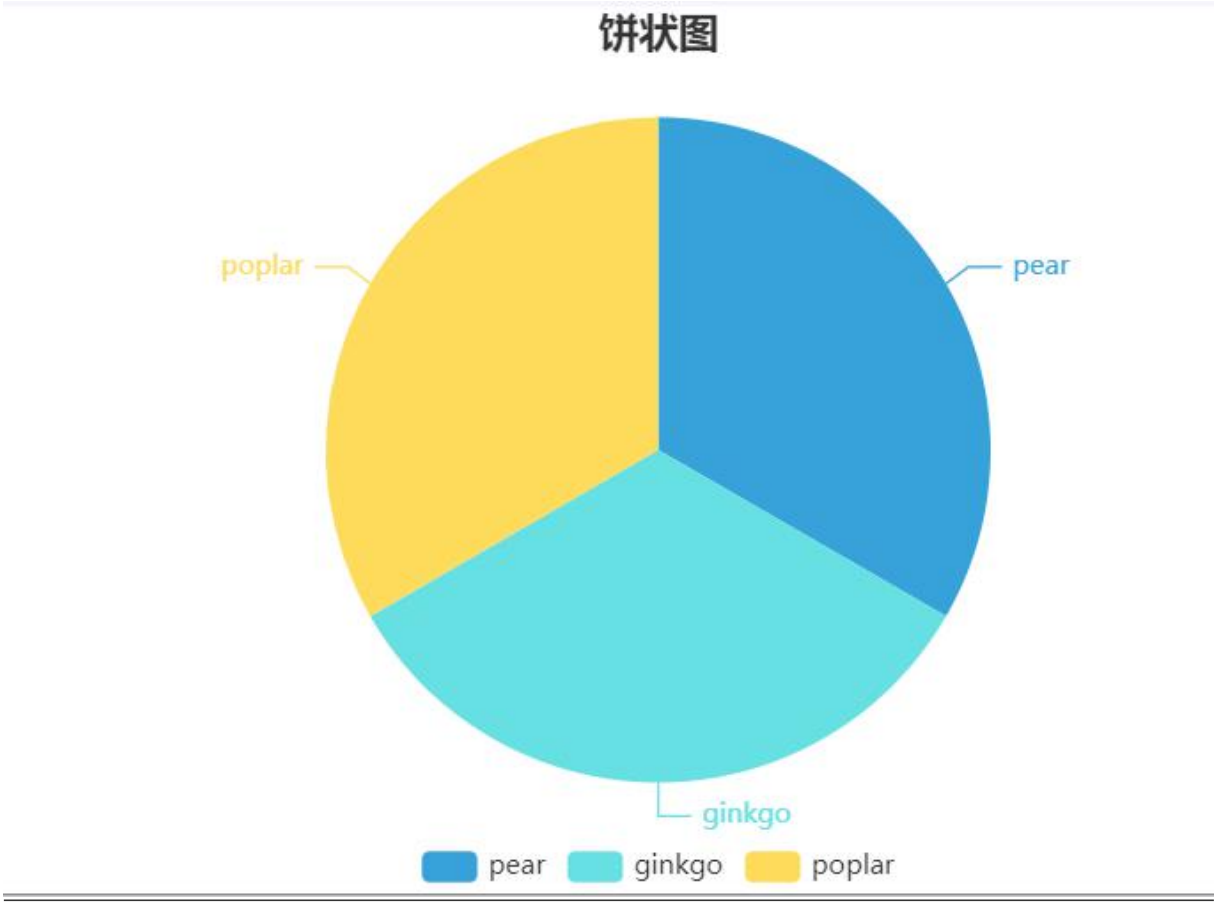
3.4.4 折线图高维数据显示



iris 数据集的折线图-图 3.4.5

每次随机选取数据集中的两个连续特征，作为 x 轴与 y 轴生成折线图，以显示种类分布受到所有不同特征的二维支配，通过低维映射高维的方式展现出联合分布。本质上是 3.4.3 散点图的升级版，除了希望得到种类与两个特征的联合分布以外，更要得出特征内潜在的变化（指上升、下降）关系。

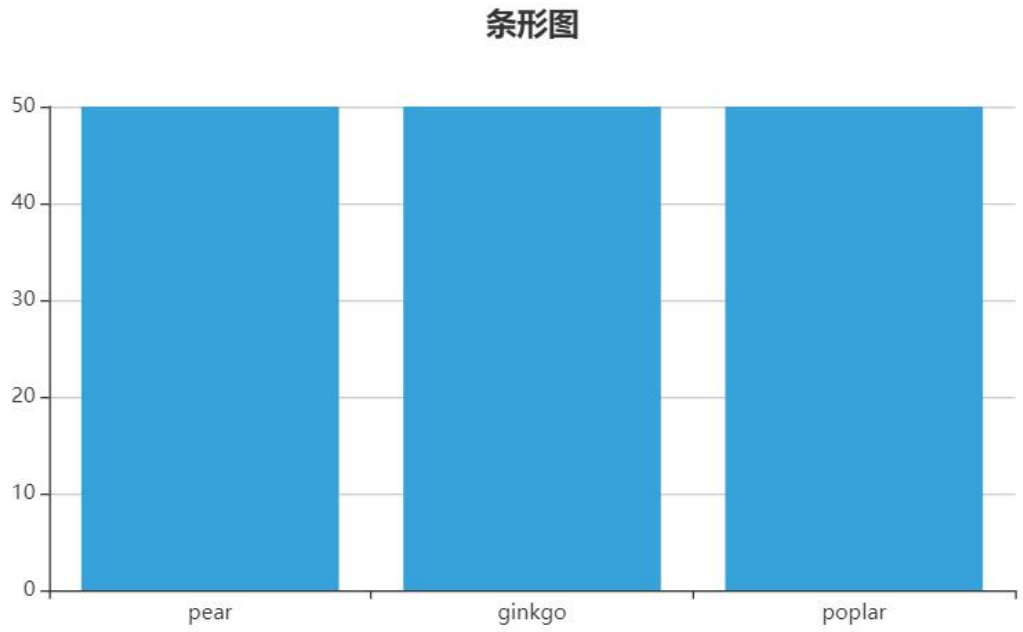
3.4.5 饼状图数据显示



iris 数据集的饼状图-图 3.4.5

显示出数据集所有的种类，以看出整个数据集的分类情况。

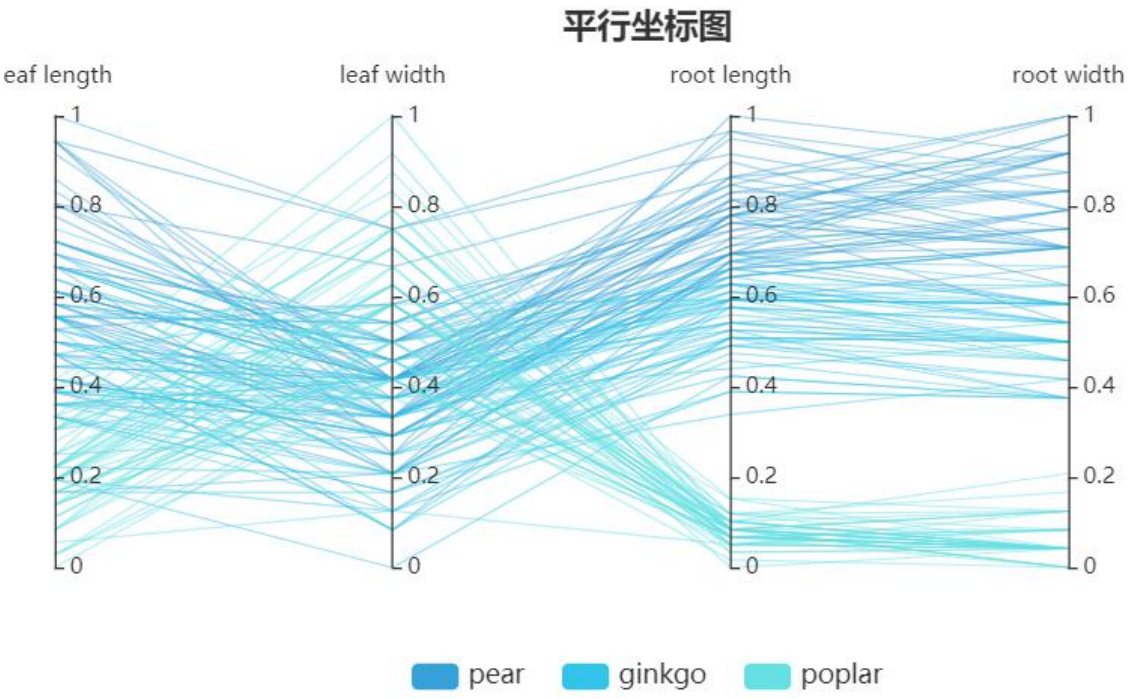
3.4.6 条形图数据显示



iris 数据集的条形图-图 3.4.6

显示出数据集所有的种类，以看出整个数据集的分类情况。

3.4.7 平行坐标图高维数据显示



iris 数据集的平行坐标图-图 3.4.6

通过与特征等同数量的垂直平行轴表示各项经过归一化的数据位置，连成直线。不

同种类的直线与直线之间的分离程度同样象征着不同种类之间的分离程度。

3.5 单一进程下的多用户调度系统

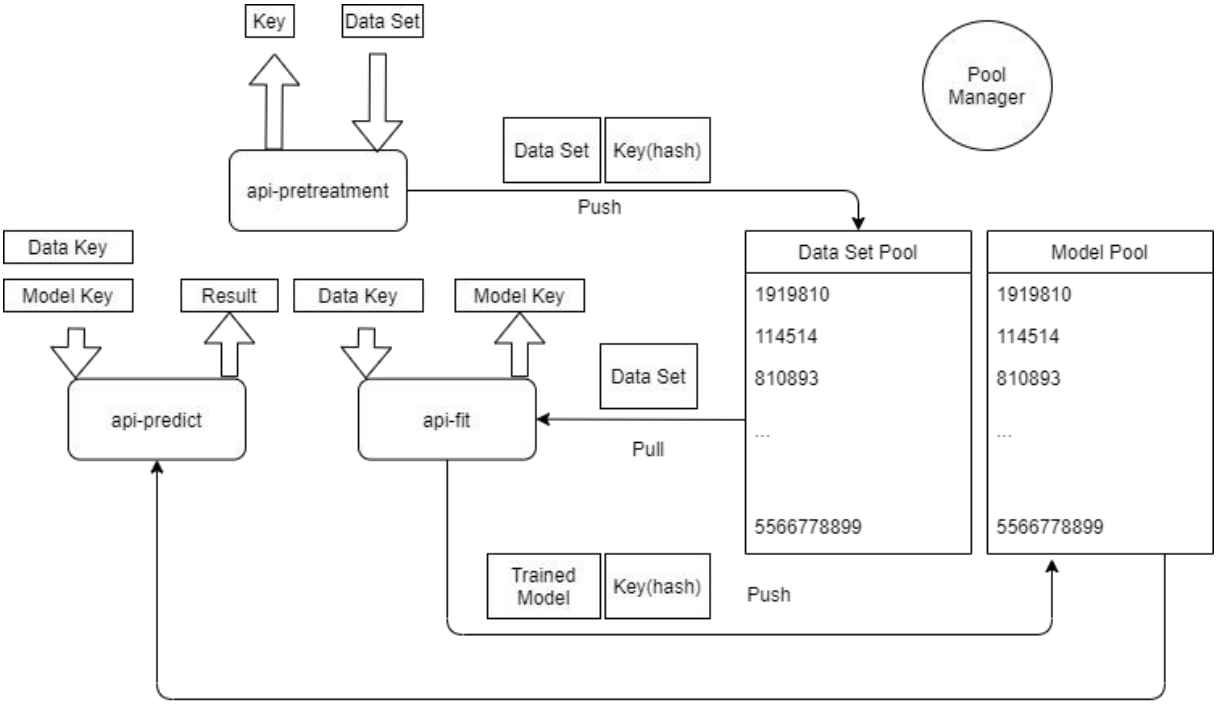
哈希表是根据设定的哈希函数 H（key）和处理冲突方法将一组关键字映射到一个有限的地址区间上，并以关键字在地址区间中的象作为记录在表中的存储位置，这种表称为哈希表或散列，所得存储位置称为哈希地址或散列地址。作为线性数据结构与表格和队列等相比，哈希表无疑是查找速度比较快的一种。

通过将单向数学函数（有时称为“哈希算法”）应用到任意数量的数据所得到的固定大小的结果。如果输入数据中有变化，则哈希也会发生变化。哈希可用于许多操作，包括身份验证和数字签名。也称为“消息摘要”。

简单解释：哈希（Hash）算法，即散列函数。它是一种单向密码体制，即它是一个从明文到密文的不可逆的映射，只有加密过程，没有解密过程。同时，哈希函数可以将任意长度的输入经过变化以后得到固定长度的输出。哈希函数的这种单向特征和输出数据长度固定的特征使得它可以生成消息或者数据。

hash 编码目的：

我们在调度系统中运用 hash 编码，将每一种算法和训练集，对应成独一无二的 hash key，从而大大加快了调度系统的效率。



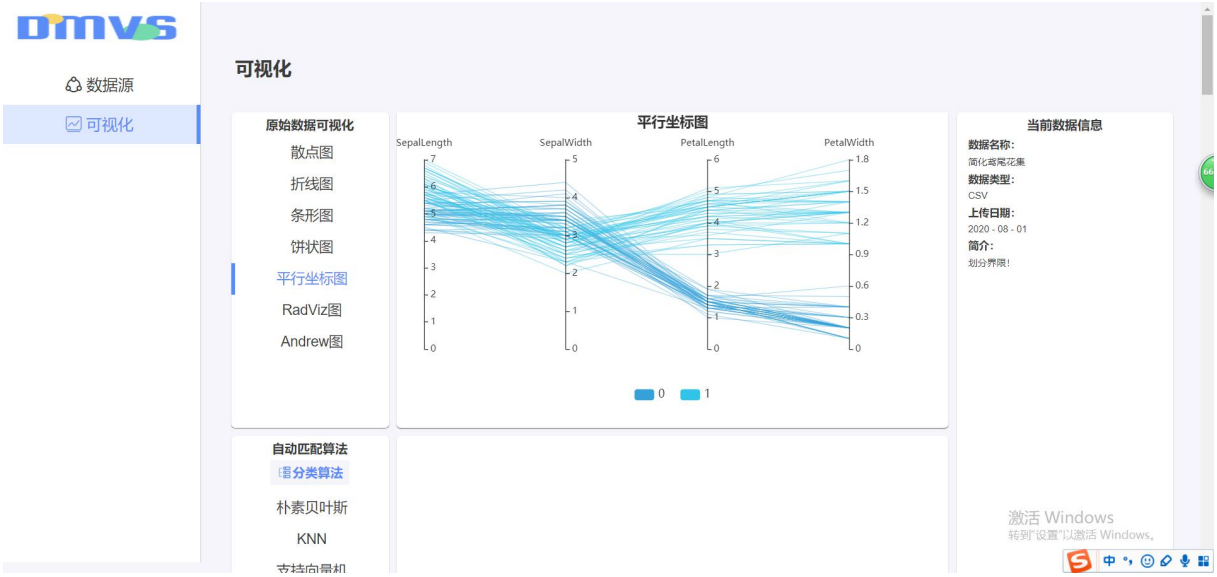
调度系统的逻辑模型-图 3.5.1

系统显然不会只受到单一用户的访问，因此在面对多用户单一进程的并发处理，我们需要确保每一个用户只能访问到自己的数据集以及训练模型，以最大程度避免数据泄漏的安全问题。

为此我们需要通过一个线性的进程去完成对于多用户的并发处理形式，因此引入了 hashKey 作为每个用户请求数据集以及模型的唯一密钥，最大程度保障用户隐私安全，同时也是调用后台中的 C/C++库进行数据的处理，加快数据处理速度。简单工作流程如图 3.5.1 所示：首先用户提交对数据进行预处理（包括去除 NaN、独热编码等）的请求，将数据集以及其他参数提交，服务器为用户返回一个独一无二的 hashKey 作为密钥，以最大程度杜绝数据劫持的发生。同时用户借助这个密钥以及选好的算法上传到服务器进行训练，训练完毕之后得到完成训练的模型的 hashKey。当用户需要进行预测的时候，只需要将数据集以及模型的密钥进行提交，再由服务器返回预测的结果并显示在前端上，仅两次的实际有效的单向数据传输一方面降低了数据的吞吐量，另一方面也降低了数据被劫持的概率。

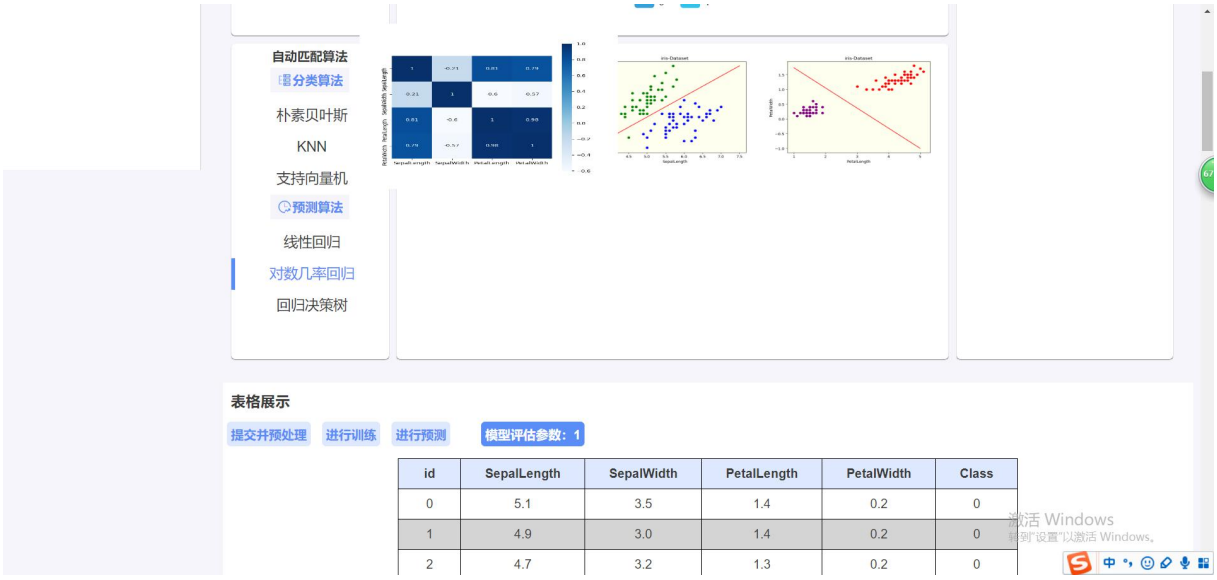
4. 系统测试

打开网站，出现的是 显示选择数据集的页面，有多种数据集供用户选择。左上角可以选择切换为可视化页面，右上角可以搜索数据源名称



在切换成可视化界面后，用户可以选择多种方式对刚才选择的数据进行可视化。

用户点击提交和预处理，和进行训练后，系统会智能选定算法对数据进行训练并建立模型。



稍等几许，便可以看到数据挖掘算法的可视化结果。

5. 结论

在这几天中，我们实现了数据挖掘可视化系统，并且实现了基本的功能，可以让用户直观的感受到可视化的原始数据以及经过数据挖掘后的数据。但是，还有一些功能，我们虽然有设想，但迫于时间原因没能够实现。由于第一次做项目，经验不足再加上线上信息沟通不便，导致团队协作十分低效。有了这一次的教训和经验，团队之间会更加磨合，更有信心面对下一次挑战。