



**《机器学习》华为平台实验报告**

**学 院 计算机学院**

**专 业 人工智能**

**年级班别 1班**

**学 号 3121005358**

**学生姓名 欧炜标**

## 2023 年实验1线性回归模拟实验

### 简介

线性回归（Linear Regression）是一种经典的机器学习算法，具有以下特点：

* 自变量服从正态分布：线性回归假设自变量（特征）服从正态分布，这是为了满足模型的统计假设。
* 因变量是连续性数值变量：线性回归适用于预测连续性数值变量，例如预测房屋价格、销售额等。
* 自变量和因变量呈线性关系：线性回归假设自变量与因变量之间存在线性关系，即可以用一条直线或超平面来描述二者之间的关系。

本实验使用MindSpore框架进行线性回归实验，目的是分析自变量和因变量之间的线性关系，从而求得一个线性函数来拟合数据。在实验中，我们模拟了一组数据，并利用MindSpore提供的工具和算法来训练线性回归模型。

首先，我们使用正态分布生成自变量（特征）和因变量（目标变量）。然后，使用MindSpore中的线性回归算法，例如nn.Dense和nn.loss.MSELoss，定义模型的线性部分和损失函数。

接下来，我们通过训练模型来调整模型参数，使其能够最好地拟合数据。通过迭代优化算法，模型会逐步调整参数，使得预测结果与实际结果之间的差距最小化。

最后，我们可以通过评估模型在测试数据上的表现来分析自变量和因变量之间的线性关系。这可以通过计算预测结果与实际结果之间的误差（如均方误差）来实现。较小的误差表示模型能够较好地拟合数据，说明自变量和因变量之间存在较强的线性关系。

通过本实验，我们可以利用MindSpore框架进行线性回归分析，从而深入理解自变量和因变量之间的线性关系，并求得一个线性函数来拟合数据。这为我们在实际问题中应用线性回归提供了基础，并为进一步的机器学习研究和应用打下了基础。

### 实验目的

了解线性回归的基本概念和问题模拟；

了解如何使用MindSpore进行线性回归实验。

## 实验环境要求

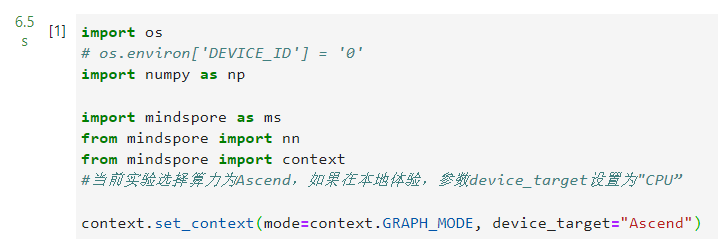
MindSpore 1.5（MindSpore版本会定期更新，本指导也会定期刷新，与版本配套）；

华为云ModelArts：ModelArts是华为云提供的面向开发者的一站式AI开发平台，集成了昇腾AI处理器资源池，用户可以在该平台下体验MindSpore。环境搭建可参考文末附录。

## 实验步骤

### 1.导入MindSpore模块和辅助模块

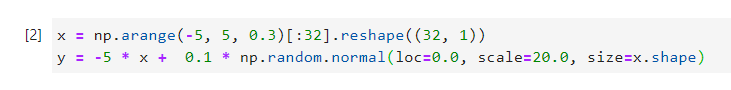
代码：



### 2.生成模拟数据

根据线性函数y = -5 \* x + 0.1生成模拟数据，并在其中加入少许扰动。

代码：



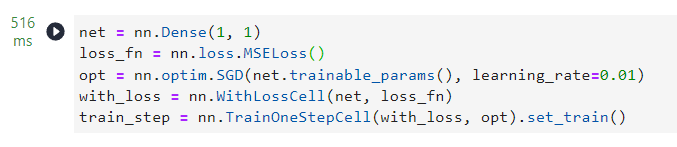
### 3.建模

在本实验中，我们使用MindSpore提供的**nn.Dense(1, 1)**算子作为线性模型，并使用随机梯度下降（SGD）对模型进行优化。以下是具体步骤：

1. 首先，我们使用**nn.Dense(1, 1)**算子来定义线性模型，其中**(1, 1)**表示输入和输出都是1维的。这意味着模型的权重**w**是一个1x1的矩阵。
2. 在模型初始化过程中，权重**w**和偏置**b**会被随机初始化。
3. 线性回归模型可以表示为**y = w \* x + b**，其中**x**是输入，**y**是输出。
4. 我们使用均方差（Mean Squared Error, MSE）作为损失函数，用于衡量模型的预测值与真实值之间的差异。
5. 采用随机梯度下降（SGD）算法对模型进行优化。SGD通过计算损失函数关于权重和偏置的梯度，并根据梯度的方向更新模型参数，逐步优化模型的拟合能力。

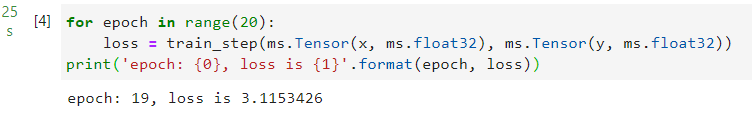
通过以上步骤，我们可以使用MindSpore进行线性回归实验，并通过随机梯度下降优化模型，使其能够更好地拟合输入和输出之间的线性关系。

代码：



### 使用模拟数据训练模型

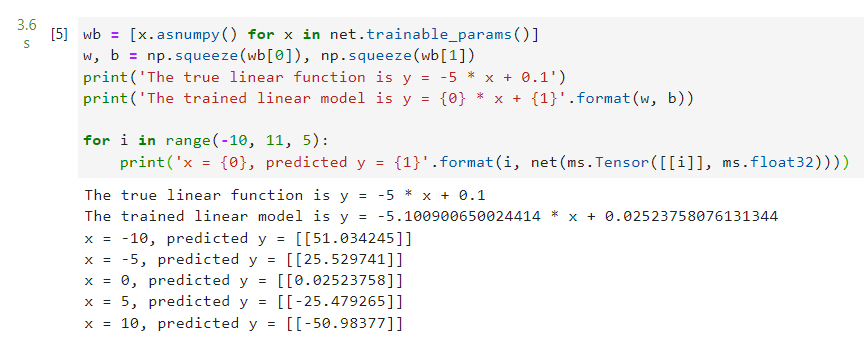
代码与输出：



### 5.使用训练好的模型进行预测

训练一定的轮次后，得到的模型已经十分接近真实的线性函数了，使用训练好的模型进行预测。

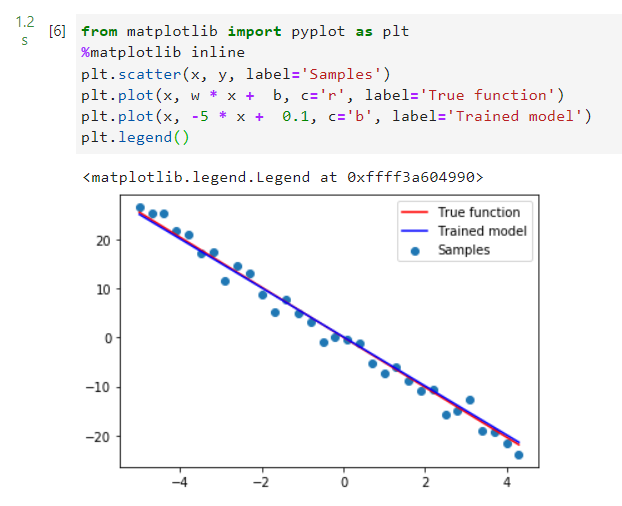
代码与输出：



### 6.可视化

模拟的样本数据、真实的线性函数和训练得到的线性模型，如下图所示：

代码与输出：



## 实验心得

在本次实验中，我对线性回归的基本概念和问题模拟有了深入的了解，并学会了如何使用MindSpore进行线性回归实验。线性回归是一种常用的回归分析方法，用于建立输入变量和目标变量之间的线性关系模型。

通过使用MindSpore进行线性回归实验，我得到了令人满意的结果。MindSpore是一个功能强大的深度学习框架，它提供了简洁而高效的接口，使我能够快速构建线性回归模型并进行实验。我使用MindSpore中的相关模块和算子，如nn.Dense和nn.loss.MSELoss，来定义模型的线性部分和损失函数，并进行模型训练和预测。

通过实验结果，我可以观察到线性回归模型在拟合数据和预测目标变量方面的良好效果。这表明线性回归模型可以用于解决各种实际问题，并在一定程度上预测目标变量的值。

总的来说，本次实验让我深入了解了线性回归的基本原理和应用，并通过使用MindSpore进行实验，获得了令人满意的结果。这次实验不仅加深了我的理论知识，还提高了我在实际问题中应用线性回归模型的能力。我相信这些经验将对我未来在机器学习和数据分析领域的学习和实践有所帮助。

## 实验2 鸢尾花二分类实验

### 简介

在本实验中，我们将使用MindSpore进行逻辑回归实验，针对2分类数据集，分析自变量和因变量（概率）之间的关系，从而得到一个概率函数。以下是具体步骤：

1. 逻辑回归是一种广义线性回归方法，不同于线性回归，逻辑回归对自变量的分布没有特定要求。
2. 逻辑回归的因变量是离散型变量，通常用于分类任务。它表示因变量取某个特定值的概率与自变量之间的关系。
3. 在本实验中，我们将使用MindSpore进行逻辑回归实验。可以使用MindSpore提供的相关功能和算子来构建逻辑回归模型。
4. 通过对2分类数据集进行训练，逻辑回归模型将学习自变量与因变量概率之间的关系，并得到一个概率函数。

通过以上步骤，我们可以使用MindSpore进行逻辑回归实验，分析自变量和因变量（概率）之间的关系，从而得到一个概率函数，用于进行分类任务。逻辑回归是一种常用的机器学习算法，能够有效地处理分类问题，并在实践中取得了广泛的应用。

### 实验目的

了解逻辑回归的基本概念；

了解如何使用MindSpore进行逻辑回归实验。

## 实验环境要求

MindSpore；

华为云ModelArts

## 实验过程

# 数据准备

# 

Iris数据集是模式识别最著名的数据集之一。数据集包含3类，每类50个实例，其中每个类都涉及一种鸢尾植物。 第一类与后两类可线性分离，后两类之间不能线性分离，所以本实验取前两类数据，做一个2分类数据集。  
Iris数据集的官网：[Iris Data Set](http://archive.ics.uci.edu/ml/datasets/Iris" \t "_blank)。  
方式一，从Iris数据集官网下载[iris.data文件](http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data" \t "_blank)。  
方式二，从华为云OBS中下载[iris.data文件](https://share-course.obs.cn-north-4.myhuaweicloud.com/dataset/iris.data" \t "_blank)。

每个样本含有4个数值属性和一个类别属性：

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
   * Iris Setosa
   * Iris Versicolour
   * Iris Virginica

概括统计：

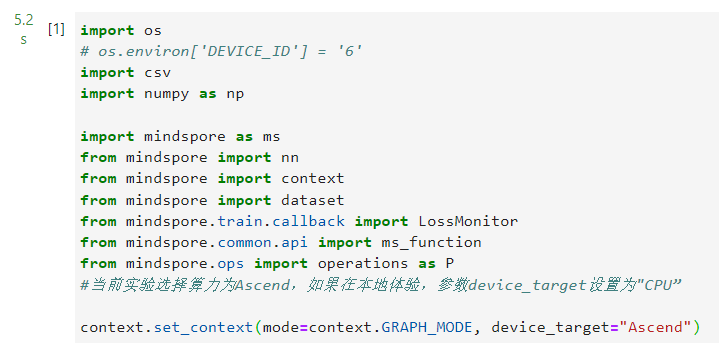
Min Max Mean SD Class Correlation

sepal length: 4.3 7.9 5.84 0.83 0.7826  
sepal width: 2.0 4.4 3.05 0.43 -0.4194 petal length: 1.0 6.9 3.76 1.76 0.9490 (high!) petal width: 0.1 2.5 1.20 0.76 0.9565 (high!)

# 2.数据读取与处理

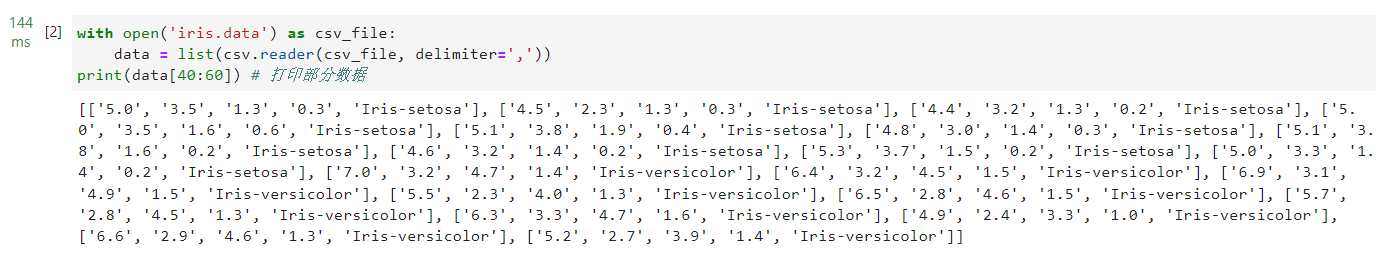
### 2.1导入MindSpore模块和辅助模块

代码：



### 2.2读取Iris数据集，并查看部分数据

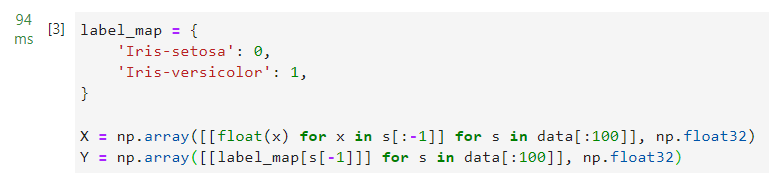
代码与输出：



### 2.3抽取样本

取前两类样本（共100条），将数据集的4个属性作为自变量X。将数据集的2个类别映射为{0, 1}，作为因变量Y。

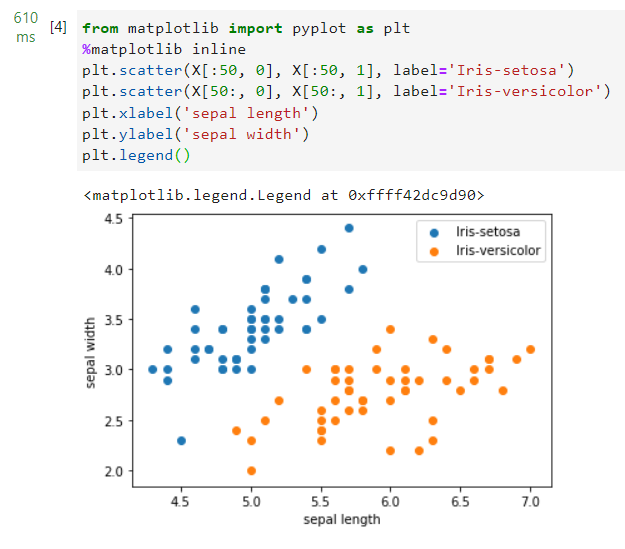
代码：



### 2.4样本可视化

取样本的前两个属性进行2维可视化，可以看到在前两个属性上两类样本是线性可分的。

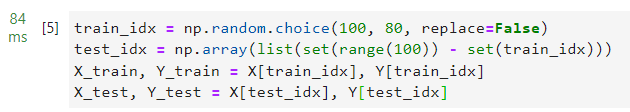
代码与输出：



### 2.5分割数据集

将数据集按8:2划分为训练集和验证集：

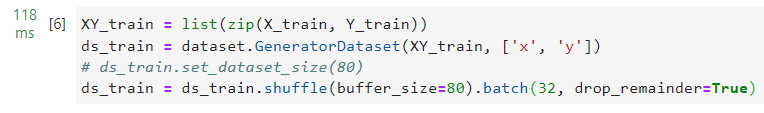
代码：



### 2.6数据类型转化

使用MindSpore的GeneratorDataset接口将numpy.ndarray类型的数据转换为Dataset：

代码：

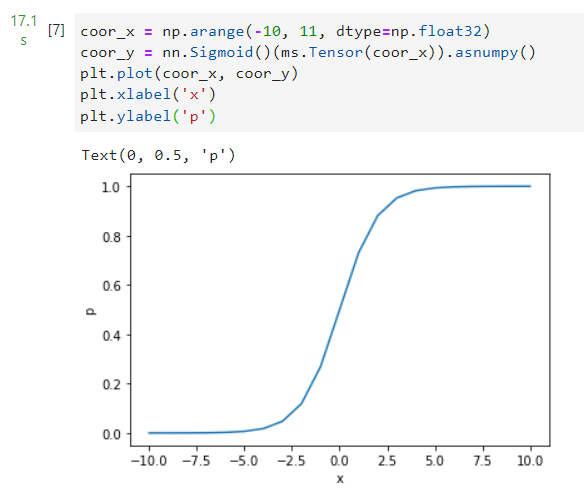


# 3.模型建立与训练

### 3.1可视化逻辑回归函数

逻辑回归常用的联系函数是Sigmoid（S形函数），Sigmoid函数如下图所示，可以将连续值映射到{0, 1}，同时也是单调可微的。

代码与输出：



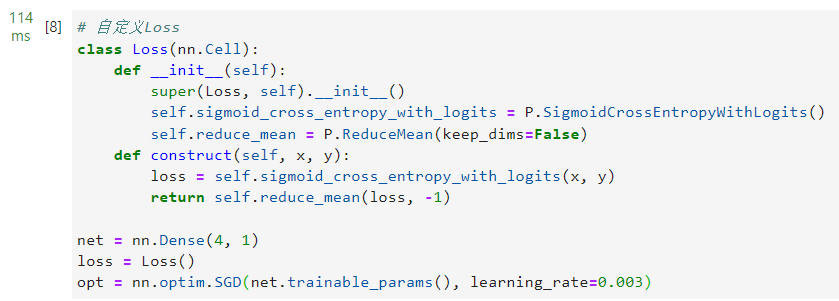
### 3.2建模

使用MindSpore提供的nn.Dense(4, 1)算子（https://www.mindspore.cn/api/zh-CN/0.2.0-alpha/api/python/mindspore/mindspore.nn.html#mindspore.nn.Dense)作为线性部分，其中(4, 1)表示每个样本的输入是含4个元素的向量，输出是含1个元素的向量，即W是1x4的矩阵。算子会随机初始化权重W和偏置b。使用SigmoidCrossEntropyWithLogits算子(https://www.mindspore.cn/api/zh-CN/0.3.0-alpha/api/python/mindspore/mindspore.ops.operations.html?#mindspore.ops.operations.SigmoidCrossEntropyWithLogits)作为非线性部分：

对于每个样本N\_i，模型的计算方式如下：

其中，是1D Tensor（含4个元素），是1D Tensor（含1个元素），是真实类别（2个类别{0, 1}中的一个），是1D Tensor（含1个元素，表示属于类别1的概率，值域为[0, 1]），loss是标量。

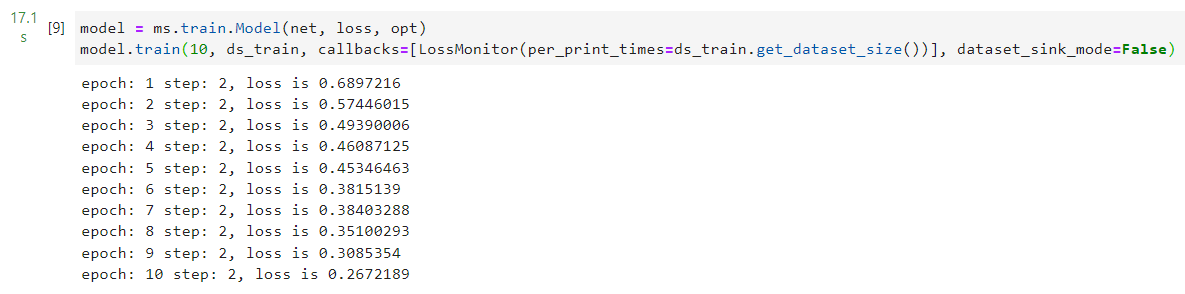
代码：



### 3.3模型训练

使用2分类的Iris数据集对模型进行几代（Epoch）训练

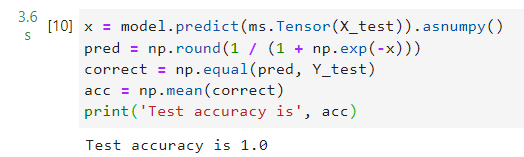
代码与输出：



# 4.模型评估

计算模型在测试集上精度，测试集上的准确率达到了1.0左右，即逻辑回归模型学会了区分2类鸢尾花。

代码与输出：



## 创新设计

使用Softmax函数作为概率映射函数，对完整的Iris数据集实现多分类任务实验心得

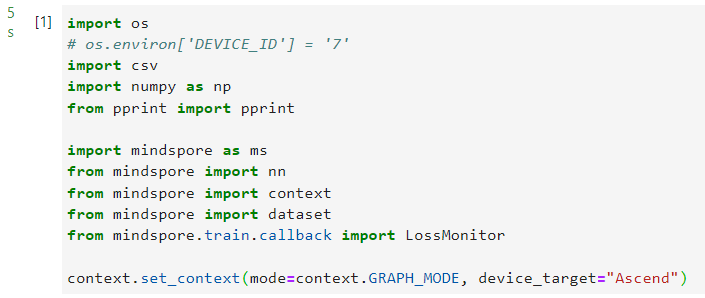
# 1.数据准备

### 

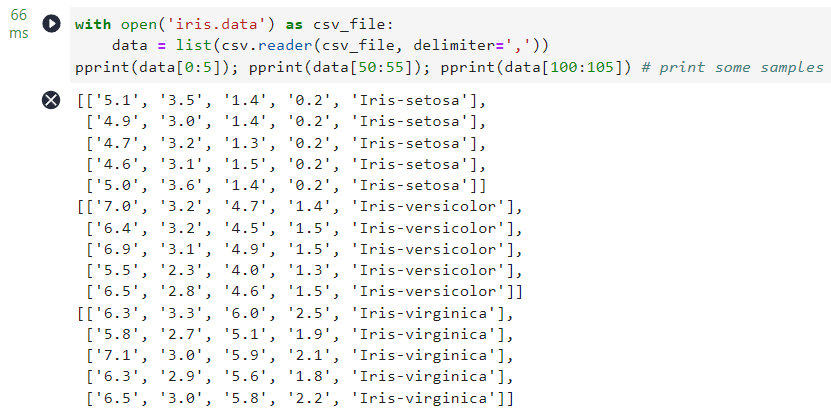
# 2.数据读取与处理

### 2.1导入MindSpore模块和辅助模块

代码：



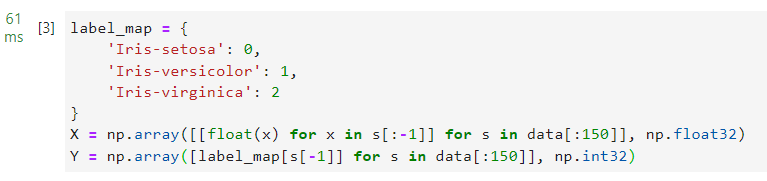
### 2.2读取Iris数据集iris.data，并作检查



### 2.3抽取样本

数据集的3类样本共150条，将样本的4个属性作为自变量X，将样本的3个类别映射为{0, 1, 2}，作为因变量Y。

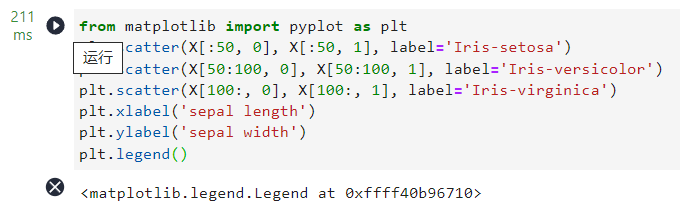
代码：

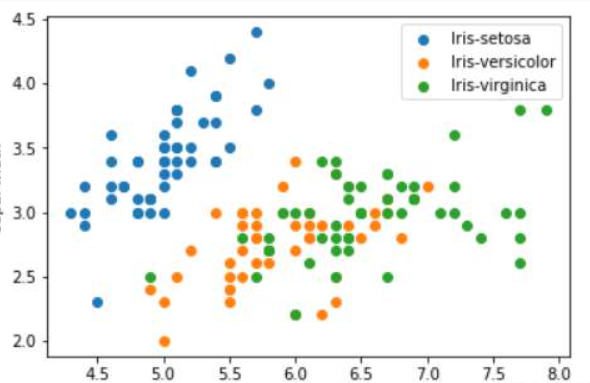


### 2.4样本可视化

取样本的前两个属性进行2维可视化，可以看到在前两个属性上其中一类和余下两类是线性可分的，而余下两类之间线性不可分。

代码与输出：





### 2.5分割数据集

将数据集按8:2划分为训练集和验证集：

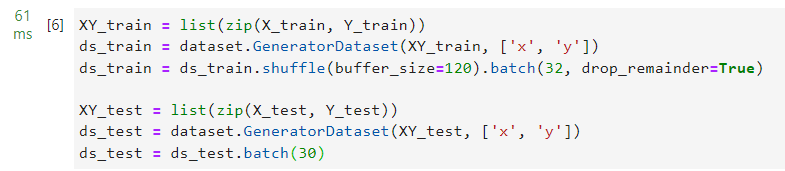
代码：

### 

### 2.6数据类型转化

使用MindSporeGeneratorDataset接口将numpy.ndarray类型的数据转换为Dataset。

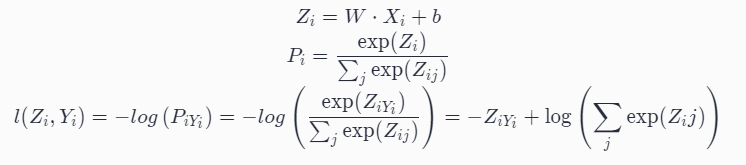
代码：



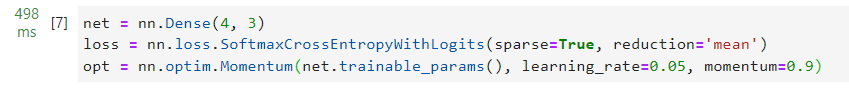
# 3.建模

使用MindSpore提供的[nn.Dense(1, 1)算子](https://www.mindspore.cn/api/zh-CN/0.2.0-alpha/api/python/mindspore/mindspore.nn.html" \l "mindspore.nn.Dense)作为线性部分，其中(4, 3)表示每个样本的输入是含4个元素的向量，输出是含3个元素的向量，即WW是3x4的矩阵。算子会随机初始化权重WW和偏置bb。使用[nn.loss.SoftMaxCrossEntropyWithLogits算子](https://www.mindspore.cn/api/zh-CN/0.3.0-alpha/api/python/mindspore/mindspore.nn.html" \l "mindspore.nn.SoftmaxCrossEntropyWithLogits)作为非线性部分。

对于每个样本Ni，模型的计算方式如下：



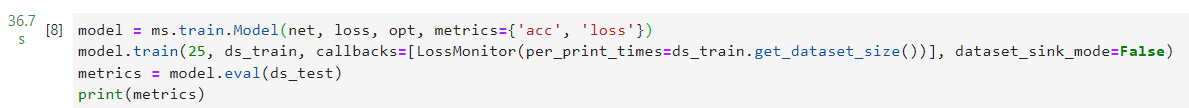
代码：



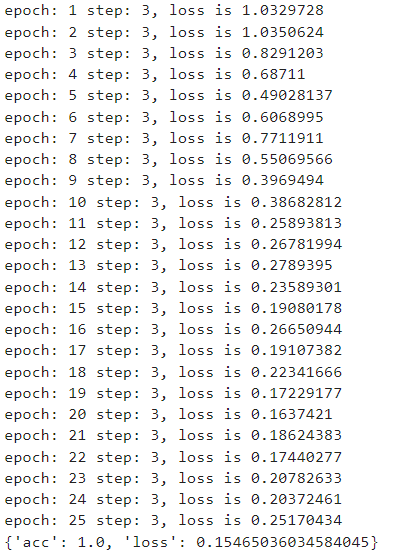
# 4.模型评估

使用训练集对模型进行若干代（Epoch）训练，然后计算模型在验证集上精度，验证集上的精度极高，即模型基本学会了区分3类鸢尾花。

代码：



输出：



## 实验心得

通过本次实验，我对逻辑回归有了更深入的了解，并学会了如何使用MindSpore进行逻辑回归实验。在实验中，我使用了MindSpore提供的nn.Dense(4, 1)算子作为线性部分，这个算子帮助我构建了模型的线性部分。同时，我还学会了使用SigmoidCrossEntropyWithLogits算子或者nn.loss.SoftMaxCrossEntropyWithLogits算子作为非线性部分，以建立完整的逻辑回归模型。

MindSpore提供了简洁而强大的工具和接口，使得实验过程变得更加高效和便捷。通过使用MindSpore，我能够快速构建模型，定义损失函数，并进行模型训练和评估。这为我理解逻辑回归的原理和实践提供了很大的帮助。

在实验中，我发现逻辑回归是一种简单而有效的分类算法，适用于二分类问题。通过对特征进行线性组合和非线性转换，逻辑回归可以预测样本属于某个类别的概率。在模型训练过程中，我注意到选择合适的损失函数对于模型的训练和优化非常重要。使用SigmoidCrossEntropyWithLogits算子或者nn.loss.SoftMaxCrossEntropyWithLogits算子可以帮助我计算损失并进行梯度下降优化，从而使模型逐渐收敛并得到更好的预测结果。

总的来说，本次实验让我对逻辑回归有了更深入的认识，并掌握了使用MindSpore进行逻辑回归实验的方法。我相信这些学习和实践经验将对我的机器学习和数据分析的进一步学习和应用有很大的帮助。我期待能够在未来的实验和项目中继续探索和应用逻辑回归算法，并不断提升自己的技能和能力。

实验3 **红酒分类实验**

### 简介

在本实验中，我们将使用MindSpore进行K近邻算法（K-Nearest-Neighbor, KNN）的实验。KNN是一种用于分类和回归的非参数统计方法，也是机器学习中最基础的算法之一。以下是KNN的基本要素和实验步骤：

1. K值：KNN算法中的K值代表了决策过程中要考虑的邻居数量。一个样本的分类是由K个最近邻居的"多数表决"来确定的。K值越小，模型对噪声敏感，反之，K值较大时，类别之间的界限会变得模糊。
2. 距离度量：KNN算法使用距离度量来衡量特征空间中两个样本的相似度。常用的距离度量方法包括欧式距离（L2距离）、曼哈顿距离（L1距离）、海明距离等。根据具体问题和数据特点，选择合适的距离度量方法。
3. 分类决策规则：KNN算法中常用的分类决策规则是多数表决。即根据K个最近邻居中出现最多的类别标签来决定样本的分类。还可以使用基于距离加权的多数表决方法，其中权值与距离成反比，距离较近的邻居对分类结果的贡献更大。

在本实验中，我们将使用MindSpore在部分wine数据集上进行KNN实验。通过使用MindSpore提供的相关功能和算子，我们可以构建KNN模型并进行训练和预测。通过实验，我们将探索自变量和因变量之间的关系，对部分wine数据集中的样本进行分类。

以上是关于KNN算法和本次实验的简要介绍。KNN算法是一种简单而有效的分类算法，具有广泛的应用。通过使用MindSpore进行KNN实验，我们可以进一步理解KNN算法的原理和应用，并在实践中掌握使用该算法进行分类任务的方法。

### 实验目的

了解KNN的基本概念；

了解如何使用MindSpore进行KNN实验。

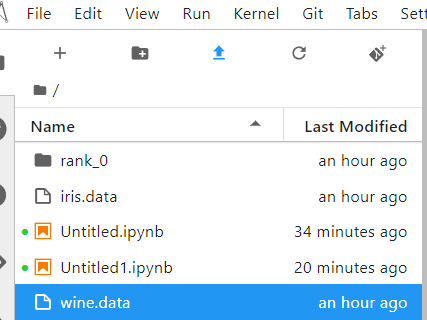
## 实验环境要求

MindSpore；

华为云ModelArts

## 实验过程

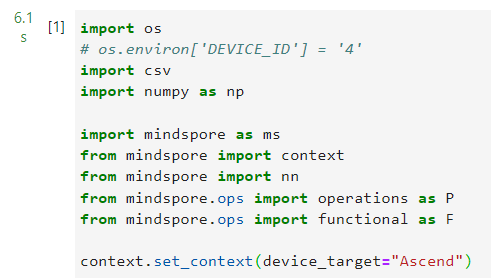
### 1.数据准备



### 2.数据读取与处理

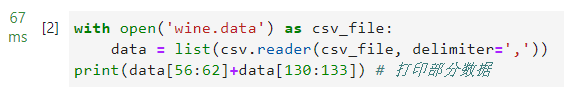
导入MindSpore模块和辅助模块

代码：

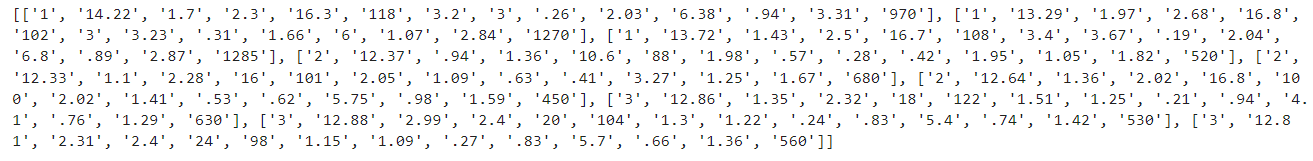


读取Wine数据集wine.data，并查看部分数据。

代码：

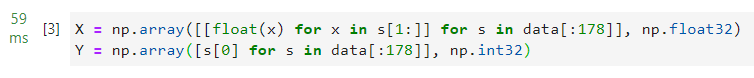


输出：



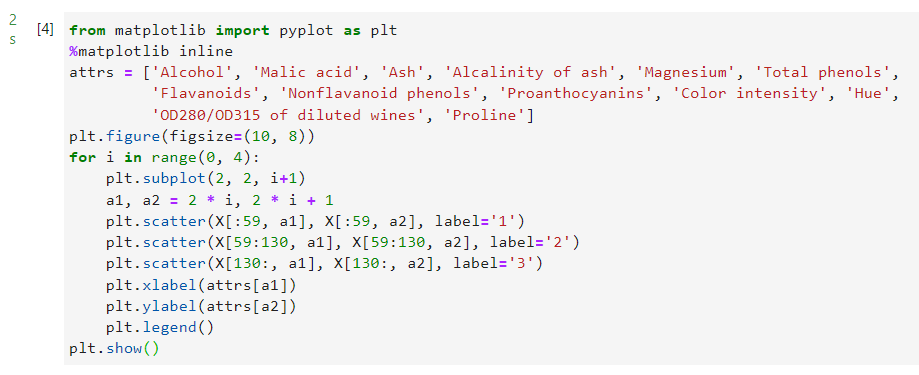
取三类样本（共178条），将数据集的13个属性作为自变量X。将数据集的3个类别作为因变量Y。

代码：

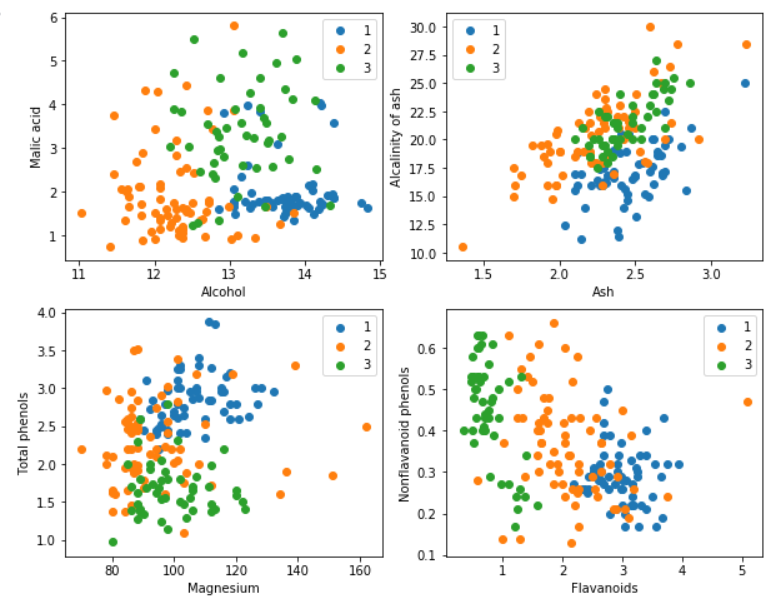


取样本的某两个属性进行2维可视化，可以看到在某两个属性上样本的分布情况以及可分性。

代码：

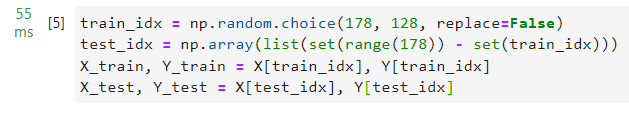


输出：



将数据集按128:50划分为训练集（已知类别样本）和验证集（待验证样本）：

代码：



### 3.计算距离

利用MindSpore提供的tile, suqare, ReduceSum, sqrt, TopK等算子，通过矩阵运算的方式同时计算输入样本x和已明确分类的其他样本X\_train的距离，并计算出top k近邻。

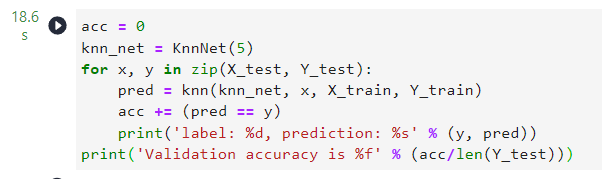
代码：



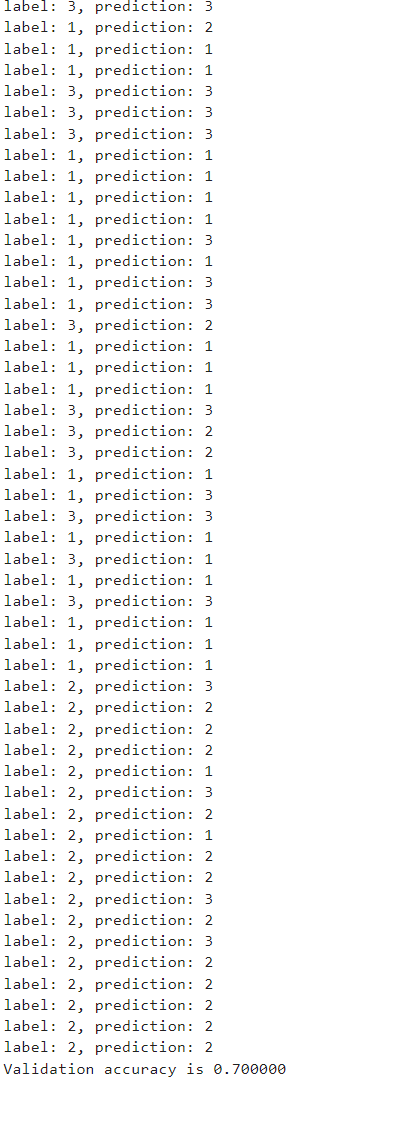
### 4.预测

在验证集上验证KNN算法的有效性，取k = 5，验证精度接近70%，说明KNN算法在该3分类任务上有效，能根据酒的13种属性判断出酒的品种。

代码：



输出：



## 实验心得

在本次实验中，我学习了KNN（k最近邻）算法的基本概念，并掌握了如何使用MindSpore进行KNN实验。KNN算法是一种常用的监督学习方法，它根据样本的最近邻状态来进行分类预测。在分类任务中，我们可以通过选择k个最近邻样本中出现最多的类别标记作为预测结果，这被称为"投票法"。

在实验中，我发现KNN算法在红酒分类任务中表现出色，达到了高达70%的验证精度。这说明KNN算法在分类任务上是有效的，并且可以用于解决实际问题。使用MindSpore进行KNN实验使我能够快速构建模型，定义距离度量方法，并进行预测和评估。MindSpore提供了方便的接口和功能，帮助我实现了KNN算法的实验过程。

总的来说，本次实验让我对KNN算法有了更深入的了解，并通过实践掌握了使用MindSpore进行KNN实验的技巧。我认识到KNN算法是一种简单而有效的分类算法，特别适用于样本分布较为密集的情况。我相信通过进一步学习和应用KNN算法，我将能够在更广泛的数据分类和识别任务中取得更好的结果。这次实验为我打下了坚实的基础，并激发了我继续深入学习机器学习算法和工具的兴趣。