

《机器学习》训练营作业（第十期）

笔记整理人：天国之影

说明

1. 每周三、周六为休息日，当天无须打卡，不会安排任何作业和任务。若学习时长中包含周三或周六，则默认忽略当天计划。
2. 本课程作业的所有代码基于 Python3.7.2，在 Jupyter Notebook 上完成。
3. 课程老师：Jian 老师
4. 电子版：周志华《机器学习》西瓜书链接：

<https://pan.baidu.com/s/1Kr-8ayfVv2fjrfz9nfaR3A>

密码：d5zm

5. 附赠：李航《统计学习方法》链接：

https://pan.baidu.com/s/1artFMZGKJ_UMd0sI488fkQ

密码：mr41

我的作业 GitHub 地址（在每一个 Week 中均有一个 MyHomeWork 文件夹，用于记录我的作业完成情况，所有 ipynb 文件均带注释）：

<https://github.com/Relph1119/MachineLearning-WatermelonBook>

1 第 1 周

1.1 学习机器学习绪论

学习时长：1 天

任务简介：

学习机器学习绪论，了解预备知识，认识群内其他小伙伴，下载书籍电子版，提前自己预习观看。

任务详解：

本部分是机器学习的绪论部分，对于这部分的学习，我并未让大家去阅读西瓜书的第一章的绪论部分，因为那部分对于初学者来讲太难了，应该就是天书，越看越蒙，这部分内容希望大家在学习完整个机器学习课程后，再回头看就好。为了让大家更轻松更好的去理解机器学习，我们特此做了这个视频，主要讲述了两个方面内容，一个是机器学习的应用案例，这部分是核心内容，另外一个机器学习理论的一个大概的分类，主要目的是想告诉同学们机器学习是什么？机器学习有哪些应用？你们应该重点关注哪些应用？希望大家在本节内容学习后，心里已经有了这三个问题的答案。

书籍下载：到公众号深度之眼后台回复“西瓜书”，即可领取电子版

视频：

机器学习绪论

打卡：

（1）内容：请给自己立一个 Flag，告诉自己未来 2 个月一定把西瓜书啃完！打完这个卡，你吹的牛逼，就会被所有人看到。所以自己吹的牛逼，含着泪也要坚持下去。

（2）形式：文字，最少 10 字；

打卡截止日期：6/19

1.2 达观杯 NLP 算法大赛

学习时长：6/20-6/21

任务简介：请按照教程讲解的内容报名，打比赛、提交比赛成绩、查看名次。并在学习群内，积极参与比赛战队的组建

任务详解：

特别注释：

第一周就打比赛，很多人会感觉懵逼，说什么都不会怎么打比赛。对的，这就是我们的目的，你先不要尝试理解，先按照老师教程，完整跑一遍比赛流程，体验一下算法的实际应用。之后在学习理论的过程中，把理论应用于比赛中，看看不同算法对比赛成绩和结果的影响，对比其中的不同，尝试提高比赛成绩。

本部分主要是给大家讲一个实际比赛的内容，这部分的主要目的是带着大家走完一个参加机器学习比赛的完整过程。对于没有参加过机器学习比赛的同学来讲，可能觉得参加比赛会很难，甚至不知如何参加，但是我想告诉你们其实很简单，希望能够带大家进入机器学习比赛这个道路上来，让大家更好的进行机器学习的实践，这也是我们的初衷，希望大家可以开启比赛之门，同时，我们提供了一个傻瓜式的完成这个比赛过程的操作文档 pdf，请大家自行下载，然后按着操作流程走完这个过程，其中会涉及比赛的报名/开发软件的安装/代码的编，写和执行/结果的提交，请大家务必完成。

PDF 下载链接：

<https://pan.baidu.com/s/1TZepfl6DXIt1cwxbf-rMQ>

提取码：[zc1e](#)

不想下载也可以直接看图文版

视频：

达观杯文本智能处理挑战赛（入门指导）

打卡：

（1）内容：提交报名成功的截图，提交比赛成绩和名次的截图

（2）形式：图片，最少 2 张；

打卡截止日期：6/21

1.3 本周学习任务简单总结

学习时长：1 天

任务简介：简单回顾本周学到几个重要知识

任务详解：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时

做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少

打卡：

（1）内容：请用文字描述，本周所学知识的重点，也可以思维导图、手写、电子版截图或者拍照均可，格式不限

（2）形式：文字，最少 20 字；图片，0-9 张；

打卡截止时间：6/23

1.4 参考答案

1. 绪论笔记

https://github.com/Relph1119/MachineLearning-WatermelonBook/blob/master/Week1/MyHomeWork/home_work_1.md

2. 1 小时打完一场比赛代码，详见 github

https://github.com/Relph1119/MachineLearning-WatermelonBook/blob/master/Week1/MyHomeWork/home_work_2.py

2 第 2 周

2.1 学习线性模型

学习时长：6/24——6/25

任务简介：

学习西瓜书 3.1/3.2/3.3，观看西瓜书公式推导学习指南和线性模型公式推导视频。

任务详解：

这部分就是我们要学习的第一个机器学习模型，线性模型，一个简单而重要的模型，前两节主要讲线性回归，这个就是我们过去学过的最小二乘法的内容，

相信大家都不陌生的。而第三小节讲的对数几率回归才是我们正章的核心，这个算法也是目前应用最广的算法之一，希望大家能够完全理清这个算法的工作流程，这也是对大家在本周最为核心的一个要求。

视频：

(1) 西瓜书公式推导——学习指南

(2) 一元线性回归公式推导

(3) 多元线性回归公式推导

(4) 对数几率回归公式推导

打卡：

(1) 内容：logistic 回归解决的是什么问题？它的表达式是什么，求导公式是什么？

(2) 形式：打卡提交总结的文字和公式推导的图片（20 字以上，1 张图片）

打卡截止日期：6/25

2.2 学习 sklearn 包中逻辑回归算法的使用

学习时长：6/27—6/28

任务简介：实验-sklearn-user guide 1.1.11

任务详解：

本节的主要任务是，了解 sklearn 包中逻辑回归算法的使用。逻辑回归算法大家基本了解了，简单使用大家也用过了。

但是这次实验的主要目的，是让大家了解一下 sklearn 包的官网，这个官网提供了详细的教程和案例，对于机器学习来讲是一种最为宝贵的资源，我们应该怎么使用这个官网来进行机器学习的学习，和实际任务的开发，这个资源也是我们日后学习的一个重点内容。

资料链接：

https://scikit-learn.org/dev/modules/linear_model.html#logistic-regression

打卡：

(1) 内容：参照链接给的代码，实际动手敲一遍，使用 `sklearn` 的逻辑回归接口实现 MNIST 分类，将运行结果截图。

https://scikit-learn.org/dev/auto_examples/linear_model/plot_sparse_logistic_regression_mnist.html#sphx-glr-auto-examples-linear-model-plot-sparse-logistic-regression-mnist-py

如果链接打不开，可直接看下方代码：

```
import time
import matplotlib.pyplot as plt
import numpy as np

from sklearn.datasets import fetch_openml
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.utils import check_random_state

print(__doc__)

# Author: Arthur Mensch <arthur.mensch@m4x.org>
# License: BSD 3 clause

# Turn down for faster convergence
t0 = time.time()
train_samples = 5000

# Load data from https://www.openml.org/d/554
X, y = fetch_openml('mnist_784', version=1, return_X_y=True)

random_state = check_random_state(0)
permutation = random_state.permutation(X.shape[0])
X = X[permutation]
y = y[permutation]
X = X.reshape((X.shape[0], -1))
```

```

X_train, X_test, y_train, y_test = train_test_split(
    x, y, train_size=train_samples, test_size=10000)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Turn up tolerance for faster convergence
clf = LogisticRegression(C=50. / train_samples,
                        multi_class='multinomial',
                        penalty='l1', solver='saga', tol=0.1)
clf.fit(X_train, y_train)
sparsity = np.mean(clf.coef_ == 0) * 100
score = clf.score(X_test, y_test)
# print('Best C % .4f' % clf.C_)
print("Sparsity with L1 penalty: %.2f%%" % sparsity)
print("Test score with L1 penalty: %.4f" % score)

coef = clf.coef_.copy()
plt.figure(figsize=(10, 5))
scale = np.abs(coef).max()
for i in range(10):
    l1_plot = plt.subplot(2, 5, i + 1)
    l1_plot.imshow(coef[i].reshape(28, 28), interpolation='nearest',
                  cmap=plt.cm.RdBu, vmin=-scale, vmax=scale)
    l1_plot.set_xticks(())
    l1_plot.set_yticks(())
    l1_plot.set_xlabel('Class %i' % i)
plt.suptitle('Classification vector for...')

run_time = time.time() - t0
print('Example run in %.3f s' % run_time)
plt.show()

```

(2) 形式：打卡提交图片，最少 1 张；

打卡截止时间：6/28

2.3 本周学习任务简单总结

学习时长：1 天

任务简介：简单回顾本周学到几个重要知识

任务详解：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少。

视频：

作业讲解-逻辑回归

打卡：

(1) 内容：请用文字描述，本周所学知识的重点，也可以思维导图、手写、电子版截图或者拍照均可，格式不限

(2) 形式：文字，最少 20 字；图片，0-9 张；

打卡截止时间：6/30

2.4 参考答案

1. **logistic** 回归解决的是什么问题？它的表达式是什么，求导公式是什么？

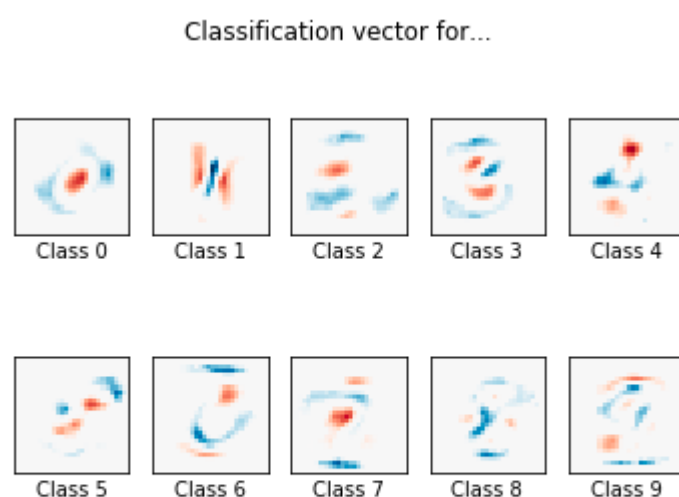
解答：

它的名字虽然是“回归”，实际解决的是分类问题。

表达式： $g(z) = \frac{1}{1+e^{-z}}$

求导公式： $g(z)(1 - g(z))$

2. 使用 **sklearn** 的逻辑回归接口实现 **MNIST** 分类，运行结果截图



```
Sparsity with L1 penalty: 81.65%  
Test score with L1 penalty: 0.8344  
Example run in 46.391 s
```

3 第3周

本周学习内容介绍：

本周我们要学习的主要内容为西瓜书的第四章决策树，在如今的机器学习领域中，树模型可以说是最为重要的模型，在各种数据挖掘的竞赛中和生产环境中，威力巨大的大杀器 **XGboost** 和 **lightGBM** 两个工具都是基于树模型构建的，所以

说大家要认真掌握本章树模型的每一个基本概念。我们需要重点把握住的一个问题就是，一棵树是怎么构建起来的？大家完成本章学习后，要做到心理能够默念构建一颗树的整个过程。

3.1 决策树的分裂准则

学习时长：7/1—7/3

任务简介：学习西瓜书 4.1/4.2，观看西瓜书决策树公式推导视频

任务详解：

这部分主要讲两方面的内容：第一，面对一个实际的数据集，应该如何构建出一颗树；第二，在构建树的过程中，树分裂节点时，如何选择出最优的属性作为分裂节点。这两方面是树模型的最为关键的部分，请大家认真搞清楚其中的每一个概念。希望大家能够掌握构建一棵树的基本流程。

视频：

决策树

打卡：

（1）内容：常见决策树算法有哪些？它们的划分准则分别是什么，是否有缺陷？

（2）形式：打卡提交总结的文字和公式推导的图片（20 字以上，1 张图片）

打卡截止日期：7/3

3.2 决策树的剪枝和连续值处理

学习时长：2 天

任务简介：阅读西瓜书 4.3/4.4

任务详解：

经过前两节的学习后，我们知道了，如何根据一个实际的数据集去构建出一颗树。但是，此时我们构建出的树可能存在一个问题，就是过拟合的问题，而解决这个问题就是 4.3 节所讲述的问题，请大家认真学习。在 4.4 节我们所接触的数据类型都是离散值，但是我们实际生产环境中，很多属性都是连续值，此时我们应该怎么处理连续数据呢，使其能够使用我们的树模型来解决问题，这就是 4.4 节要讲述的问题之一。同时 4.4 节还讲了一个重要问题，是我们生产环境经常面

临的问题，就是当我们采集一些数据的时候，经常有一些数据的某些值是空的，这就严重影响树的构建，那么应该怎么解决呢？这就是 4.4 节需要讲述的另一个问题。简而言之，请大家掌握两个问题：第一，如何进行树的剪枝来防止过拟合，第二，对于含有空值的数据，此时应该怎么构建树。

打卡：

（1）内容：决策树为什么要剪枝？有几种方法？简述一下，并分析其优缺点

（2）形式：文字，最少 50 字；

打卡截止日期：7/5

3.3 学习 sklearn 包中决策树算法的使用

学习时长：1 天

任务简介：实验-sklearn-user guide 1.10

任务详解：

了解 sklearn 包中的决策树算法是如何使用的，请大家认真看完 1.10.1 的内容，重点关注两部分内容：第一，怎么使用“DecisionTreeClassifier”这个 api，第二，这个 api 中的参数是什么意思。其中在最末端有一个案例的链接，大家可选择看，不作为任务要求。同学们，对这部分学习的深入程度，依据个人代码能力而定。

资料链接：<https://scikit-learn.org/dev/modules/tree.html>

打卡：

（1）内容：参照链接给的代码或者下图代码，实际动手敲一遍，使用 sklearn 的决策树接口实现鸢尾花分类，将运行结果截图。

https://scikit-learn.org/dev/auto_examples/tree/plot_iris_dtc.html#sphx-glr-auto-examples-tree-plot-iris-dtc-py

如果链接打不开，可直接看下方代码：

```

print(__doc__)

import numpy as np
import matplotlib.pyplot as plt

from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier, plot_tree

# Parameters
n_classes = 3
plot_colors = "ryb"
plot_step = 0.02

# Load data
iris = load_iris()

for pairidx, pair in enumerate([[0, 1], [0, 2], [0, 3],
                                [1, 2], [1, 3], [2, 3]]):
    # We only take the two corresponding features
    X = iris.data[:, pair]
    y = iris.target

    # Train
    clf = DecisionTreeClassifier().fit(X, y)

    # Plot the decision boundary
    plt.subplot(2, 3, pairidx + 1)

    x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, plot_step),
                          np.arange(y_min, y_max, plot_step))
    plt.tight_layout(h_pad=0.5, w_pad=0.5, pad=2.5)

    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    cs = plt.contourf(xx, yy, Z, cmap=plt.cm.RdYlBu)

    plt.xlabel(iris.feature_names[pair[0]])
    plt.ylabel(iris.feature_names[pair[1]])

    # Plot the training points
    for i, color in zip(range(n_classes), plot_colors):
        idx = np.where(y == i)
        plt.scatter(X[idx, 0], X[idx, 1], c=color, label=iris.target_names[i],
                    cmap=plt.cm.RdYlBu, edgecolor='black', s=15)

plt.suptitle("Decision surface of a decision tree using paired features")
plt.legend(loc='lower right', borderpad=0, handletextpad=0)
plt.axis("tight")

plt.figure()
clf = DecisionTreeClassifier().fit(iris.data, iris.target)
plot_tree(clf, filled=True)
plt.show()

```

(2) 形式：打卡提交图片，最少 1 张；

打卡截止时间：7/7

3.4 本周学习任务简单总结

学习时长：1 天

任务简介：简单回顾本周学到几个重要知识

任务详解：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少。

视频：

作业讲解-决策树

打卡：

(1) 内容：请用文字描述，本周所学知识的重点，也可以思维导图、手写、电子版截图或者拍照均可，格式不限

(2) 形式：文字，最少 20 字；图片，0-9 张；

打卡截止时间：7/7

3.5 参考答案

1. 常见决策树算法有哪些？它们的划分准则分别是什么，是否有缺陷？

解答：

常见决策树算法有：ID3、C4.5、CART

划分准则分别是：信息增益、信息增益率、基尼指数

信息增益缺点：对可取值较多的属性有偏好，比如编号、日期

信息增益率缺点：对可取值较少的属性有偏好

2. 决策树为什么要剪枝？有几种方法？简述一下，并分析其优缺点

解答：

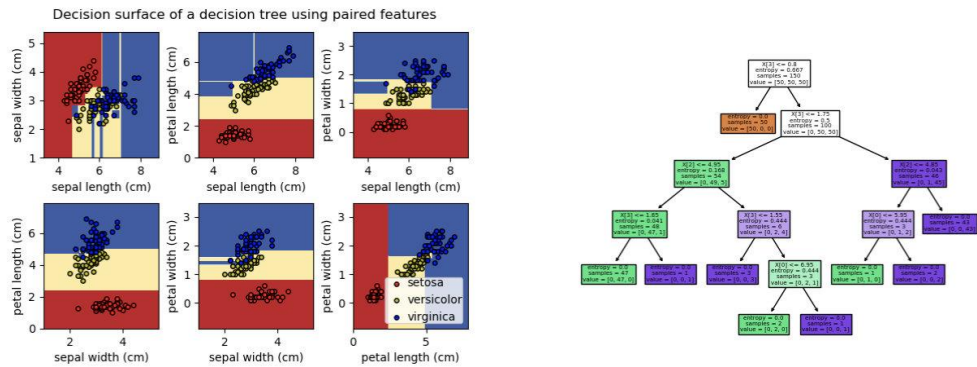
为了防止过拟合。有两种方法：预剪枝、后剪枝。

- 预剪枝：预剪枝的核心思想是在树中进行扩展之前，先计算当前的划分是否能带来模型泛化能力的提升，如果不能则不再生长子树。此时可能存在不同

类别的样本同时存在结点中，按照多数投票原则判断该结点所属类别。优点：思想直接、算法简单、效率高等特点，适合解决大规模问题。缺点：有欠拟合风险。

- 后剪枝：后剪枝的核心思想是让算法生成一棵完全生长的决策树，然后从最底层向上计算是否剪枝。剪枝过程将子树删除，用一个叶子结点代替，该结点的类别同样按照多数投票的原则进行判断。同样，后剪枝也可以通过在测试集的准确率进行判断，如果剪枝后准确率有提升则进行剪枝。优点：相比于预剪枝，泛化能力强。缺点：时间开销大

3. 使用 sklearn 的决策树接口实现鸢尾花分类，运行结果截图



4 第 4 周

本周学习内容介绍：

本周我们要学习的主要内容为西瓜书的第 6 章支持向量机，在深度学习未火之前，支持向量机一直都是学术界和工业界的热点，这得益于其优秀的性能，特别是面对数据规模比较小的时候，不同简单的逻辑回归，支持向量机能够构建出数据之间的非线性关系。同时，在机器学习的面试过程中，支持向量机一直都是热点问题，因为这部分能够对同学的数学功底进行考察，所以请大家认真对待这章的关键公式。学习完本章后，希望大家能够理清两个逻辑思维：一个是，支持向量机的原始公式是怎么由实际问题产生，这涉及到的是算法灵魂部分，希望大家认真思考；第二个就是数学推导，原始公式如何推导至其对偶形式的。

4.1 支持向量机原始模型的建立和求解

学习时长：7/8——7/9

任务简介：学习西瓜书 6.1/6.2，观看西瓜书的支持向量机公式推导视频

任务详解：

这部分主要讲两方面的内容：第一，支持向量机的原始公式是怎么由实际问题产生，这涉及的是算法的思想，也就是数学模型的建立；第二，当将数学模型建立后，接下来要做的就是对数学模型进行求解，这是这部分的主要内容，请大家掌握原始问题到对偶问题的数学推导过程。这部分的学习推导，大家可重点学习下李航老师的《统计学习方法》的内容，如果大家觉得公式推导吃力，推荐大家学习的《统计学习方法》训练营。

视频：

- (1) SVM 原始形式公式推导
- (2) SVM 对偶形式公式推导
- (3) SVM 问题的求解算法 (SMO)

打卡：

- (1) 内容：谈谈 SVM 原理是什么？
- (2) 形式：打卡提交总结的文字和公式推导的图片（20 字以上，1 张图片）

打卡截止日期：7/9

4.2 核函数和软间隔支持向量机

学习时长：7/11—7/12

任务简介：学习西瓜书 6.3/6.4/6.5，观看西瓜书公式推导视频

任务详解：

经过前两节的学习后，我们知道了支持向量机的原理，以及如何求解支持向量机问题。前两节讲述的内容都是建立在数据是线性可分的情况下进行介绍的，然而在我们的实际工程场景中，我们的数据一般都是线性不可分，那么我们应该如何用前两节学习过的支持向量机来解决这个问题呢？这其中所涉及的一个重要技术就是将线性不可分的数据映射到更高维的空间中，使其变成线性可分的数据，这就是 6.3 节中主要介绍的内容。虽然我们可以在理论上使用一个函数将原线性不可分的数据变成完全线性可分的数据，但实际工程中，很难将其映射成完

全线性可分，为了解决这个问题，同时增加模型的泛化能力，这就是 6.4 节所要讲述的解决方案。这部分的学习推导，大家可重点学习下李航老师的《统计学习方法》的内容。

视频：

（1）Soft-Margin SVM 公式推导

（2）Support Vector Regression（SVR）公式推导

打卡：

（1）内容：为什么 SVM 要引入核函数？

（2）形式：打卡提交总结的文字和公式推导的图片（20 字以上，1 张图片）

打卡截止日期：7/12

4.3 了解 sklearn 包中 svm 算法的使用

学习时长：7/13—7/14

任务简介：阅读 sklearn-user guide 1.4.1

任务详解：

了解 sklearn 包中的 SVM 算法是如何使用的，请大家认真看完 1.4.1 的内容，重点关注两部分内容：第一，怎么使用 SVC 这个 API，第二，这个 API 中的参数是什么意思。其中在最末端有一个案例的链接，大家可选择看，不作为任务要求。同学们，对这部分学习的深入程度，依据个人代码能力而定。

资料链接：<https://scikit-learn.org/dev/modules/svm.html>

打卡：

（1）内容：参照链接给的代码或者下图代码，实际动手敲一遍，使用 sklearn 的 SVM 接口实现鸢尾花分类，将运行结果截图。

https://scikit-learn.org/dev/auto_examples/svm/plot_iris_svc.html#sphx-glr-auto-examples-svm-plot-iris-svc-py

如果链接打不开，可直接看下方代码：

```

print(__doc__)

import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm, datasets

def make_meshgrid(x, y, h=.02):
    """Create a mesh of points to plot in

    Parameters
    -----
    x: data to base x-axis meshgrid on
    y: data to base y-axis meshgrid on
    h: stepsize for meshgrid, optional

    Returns
    -----
    xx, yy : ndarray
    """
    x_min, x_max = x.min() - 1, x.max() + 1
    y_min, y_max = y.min() - 1, y.max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                          np.arange(y_min, y_max, h))
    return xx, yy

```



```

def plot_contours(ax, clf, xx, yy, **params):
    """Plot the decision boundaries for a classifier.

    Parameters
    -----
    ax: matplotlib axes object
    clf: a classifier
    xx: meshgrid ndarray
    yy: meshgrid ndarray
    params: dictionary of params to pass to contourf, optional
    """
    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    out = ax.contourf(xx, yy, Z, **params)
    return out

# import some data to play with
iris = datasets.load_iris()
# Take the first two features. We could avoid this by using a two-dim dataset
X = iris.data[:, :2]
y = iris.target

# we create an instance of SVM and fit out data. We do not scale our
# data since we want to plot the support vectors
C = 1.0 # SVM regularization parameter
models = (svm.SVC(kernel='linear', C=C),
          svm.LinearSVC(C=C, max_iter=10000),
          svm.SVC(kernel='rbf', gamma=0.7, C=C),
          svm.SVC(kernel='poly', degree=3, gamma='auto', C=C))
models = (clf.fit(X, y) for clf in models)

```

```

# title for the plots
titles = ('SVC with linear kernel',
          'LinearSVC (linear kernel)',
          'SVC with RBF kernel',
          'SVC with polynomial (degree 3) kernel')

# Set-up 2x2 grid for plotting.
fig, sub = plt.subplots(2, 2)
plt.subplots_adjust(wspace=0.4, hspace=0.4)

X0, X1 = X[:, 0], X[:, 1]
xx, yy = make_meshgrid(X0, X1)

for clf, title, ax in zip(models, titles, sub.flatten()):
    plot_contours(ax, clf, xx, yy,
                  cmap=plt.cm.coolwarm, alpha=0.8)
    ax.scatter(X0, X1, c=y, cmap=plt.cm.coolwarm, s=20, edgecolors='k')
    ax.set_xlim(xx.min(), xx.max())
    ax.set_ylim(yy.min(), yy.max())
    ax.set_xlabel('Sepal length')
    ax.set_ylabel('Sepal width')
    ax.set_xticks(())
    ax.set_yticks(())
    ax.set_title(title)

plt.show()

```

(2) 形式：打卡提交图片，最少 1 张；

打卡截止时间：7/14

4.4 本周学习任务简单总结

学习时长：1 天

任务简介：简单回顾本周学到几个重要知识以及观看作业讲解视频

任务详解：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少。

视频：

作业讲解-SVM

打卡：

(1) 内容：请用文字描述，本周所学知识的重点，也可以思维导图、手写、电子版截图或者拍照均可，格式不限

(2) 形式：文字，最少 20 字；图片，0-9 张；

4.5 参考答案

1. 谈谈 SVM 原理是什么？

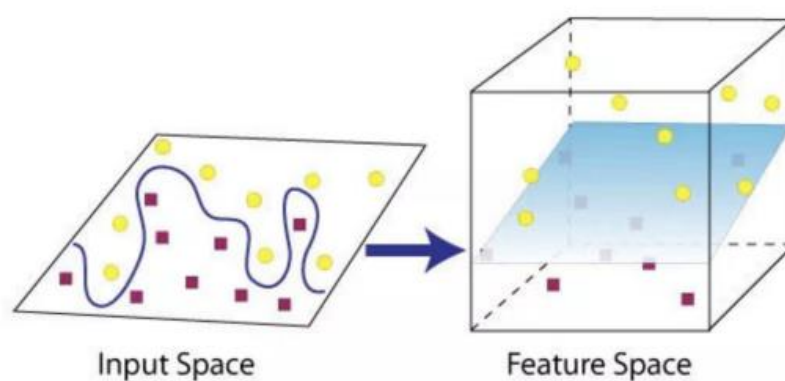
解答：

SVM 是一种二类分类模型。它的基本模型是在特征空间中寻找间隔最大化的分离超平面的线性分类器。

- 当训练样本线性可分时，通过硬间隔最大化，学习一个线性分类器，即线性可分支持向量机；
- 当训练数据近似线性可分时，引入松弛变量，通过软间隔最大化，学习一个线性分类器，即线性支持向量机；
- 当训练数据线性不可分时，通过使用核技巧及软间隔最大化，学习非线性支持向量机。

以上各种情况下的数学推到应当掌握，硬间隔最大化（几何间隔）、学习的对偶问题、软间隔最大化（引入松弛变量）、非线性支持向量机（核技巧）。

2. 为什么 SVM 要引入核函数？

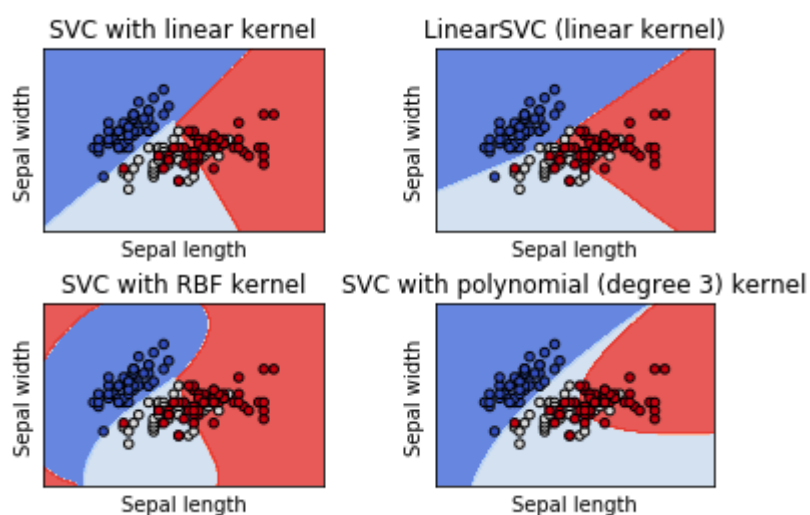


解答：

当样本在原始空间线性不可分时，可将样本从原始空间映射到一个更高维的特征空间，使得样本在这个特征空间内线性可分。而引入这样的映射后，所求解的对偶问题的求解中，无需求解真正的映射函数，而只需要知道其核函数。核函数的定义： $K(x, y) = \langle \varphi(x), \varphi(y) \rangle$ ，即在特征空间的内积等于它们在原始样本空间中通过核函数 K 计算的结果。一方面数据变成了高维空间中线性可分的数

据，另一方面不要求解具体的映射函数，只需要给定具体的核函数即可，这样使得求解的难度大大降低。

3. 使用 `sklearn` 的 `SVM` 接口实现鸢尾花分类，将运行结果截图



5 第 5 周

本周学习内容介绍：

本周我们要学习的主要内容为西瓜书的第 7 章贝叶斯分类器。本章最为重要的内容就是极大似然估计和朴素贝叶斯分类器。其中极大似然法贯穿我们整个机器学习和深度学习的理论中，是很多学习算法的基础，所以大家比如掌握。另外朴素贝叶斯，因为其优秀的速度优势，比如过去的判别邮件是否是垃圾邮件的任务中，一直扮演着重要决策，所以大家应该了解其思想。7.1 节的内容太过于抽象，无法理解是很正常的，同学们不要纠结，直接放弃就好，后续的章节才是重点。

这周的内容会简单一些，因为大家经历了 `SVM` 的折磨，希望大家可以这周轻松一些，同时也可以对以前没有掌握好的的内容，进行重点补充学习。

5.1 极大似然估计与朴素贝叶斯

学习时长：7/15——7/16

任务简介：学习西瓜书 7.2/7.3，观看极大似然估计与朴素贝叶斯视频

任务详解：

这部分讲的最为核心的一个内容就是极大似然估计，极大似然法就是如何根据可获得的一些样本去估计出总体分布的方法。极大似然估计是概率统计学中的内容，同时这个方法一直贯穿我们机器学习和深度学习的理论中，是很多算法的基础，请大家牢牢掌握其解决问题的背后思想，大家必须对这个算法思想有直觉性的了解。

视频：

朴素贝叶斯公式推导

打卡：

（1）内容：朴素贝叶斯分类器为什么引入拉普拉斯平滑？（作业答案以及讲解视频会在周日公布）

（2）形式：文字，至少 10 个字。

打卡截止日期：7/16

5.2 EM 算法

学习时长：7/18——7/19

任务简介：学习西瓜书 7.6，观看 EM 算法视频

任务详解：

前面讲述的章节，我们都是假设训练样本的所有属性的值都是已知，但是往往实际工程中，有些属性值是观测不到的，即未知的。那么我们应该对模型的参数进行估计的，这就是我们这节的 EM 算法的内容。这个算法因为属于统计学的内容，所以比较难于理解，但是这个算法曾经入选数据挖掘的十大算法，希望同学们可以理解其算法思路。部分的学习推导，大家可重点学习下李航老师的《统计学习方法》的内容。

视频：

（1）EM 算法的引入和例子

（2）EM 算法的导出

（3）EM 算法的求解例子

打卡：

(1) 内容：什么时候使用 EM 算法？E 和 M 分别做什么？

(2) 形式：文字，至少 20 字。

打卡截止日期：7/19

5.3 了解 sklearn 包中的朴素贝叶斯算法的适用

学习时长：7/20——7/21

任务简介：阅读 sklearn-user guide 1.9.1/1.9.2/1.9.3

任务详解：

了解 sklearn 包中的朴素贝叶斯算法是如何使用的，请大家认真看完相关内容，重点关注一个核心内容，高斯分布贝叶斯/多项式贝叶斯/伯努利贝叶斯的区别和应用场景。

资料链接：

http://scikit-learn.org/stable/modules/naive_bayes.html

打卡：

(1) 内容：参照链接给的代码或者下图代码，实际动手敲一遍，使用 sklearn 的高斯贝叶斯接口实现鸢尾花分类，将运行结果截图。

http://scikit-learn.org/stable/modules/naive_bayes.html

```
from sklearn import datasets
iris = datasets.load_iris()
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
y_pred = gnb.fit(iris.data, iris.target).predict(iris.data)
print("Number of mislabeled points out of a total %d points : %d"
      % (iris.data.shape[0], (iris.target != y_pred).sum()))
```

(2) 形式：图片，至少 1 张；

打卡截止时间：7/21

5.4 本周学习任务简单总结

学习时长：1 天

任务简介：简单回顾本周学到几个重要知识以及观看作业讲解视频

任务详解：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少。

视频：

作业讲解-贝叶斯和 EM 算法

打卡：

(1) 内容：请用文字描述，本周所学知识的重点，也可以思维导图、手写、电子版截图或者拍照均可，格式不限

(2) 形式：文字，最少 20 字；图片，0-9 张；

5.5 参考答案

1. 朴素贝叶斯分类器为什么引入拉普拉斯平滑？

解答：为了防止条件概率 $P(X|Y)$ 出现概率为 0。

2. 什么时候使用 EM 算法？E 和 M 分别做什么？

解答：

如果概率模型的变量都是观测变量，则给定数据之后，可以直接用极大似然估计法或者贝叶斯估计法来估计模型参数。

但是当模型含有隐变量时，就不能简单的使用这些估计方法。此时需要使用 EM 算法。

- EM 算法是一种迭代算法。
- EM 算法专门用于含有隐变量的概率模型参数的极大似然估计，或者极大后验概率估计。

EM 算法的每次迭代由两步组成：

- E 步求期望。
- M 步求极大。

所以 EM 算法也称为期望极大算法。

3. 使用 sklearn 的高斯贝叶斯接口实现鸢尾花分类，将运行结果截图

Number of mislabeled points out of a total 150 points : 6

6 第6周

本周学习内容介绍：

本周我们要学习的主要内容为西瓜书的第5章神经网络。本节的内容我觉得对于大多数同学来讲，是最为重要的一章，没有之一！为什么这么说呢？因为目前最火的深度学习就泛指用神经网络搭建的深层模型，因为深度学习在自然语言处理和计算机视觉领域，相比传统的机器学习算法来说，占有了绝对的优势，得益于其优秀的拟合能力，深度学习也是做机器学习人必须要会的一项技能，而本章又是深度学习的基础，所以希望同学们认真学习本章，特别是对本章数学公式的推导要完全掌握！因为这部分公式推导是目前整个深度学习算法的核心和基础！

6.1 神经网络结构

学习时长：7/22——7/23

任务简介：学习西瓜书 5.1/5.2

任务详解：

这部分主要介绍神经网络的组成结构，即都是由哪些单元搭建出来的，以及每个单元的作用是什么。这部分内容比较简单，希望大家可以清晰的知道一个过程，就是数据进行网络，都对数据进行了哪些运算，直至输出出来。如果大家觉得学的不是很懂，建议大家读读《深度学习》花书。

视频：

神经网络公式推导

打卡：

（1）内容：课后习题 5.1 试述将线性函数 $f(x) = w^t x$ 用作神经元激活函数的缺陷和 5.2 试述使用图 5.2(b)激活函数的神经元与对率回归的联系。

（2）形式：文字，至少 30 字

打卡截止日期：7/23

6.2 BP 算法

学习时长：7/23——7/24

任务简介：学习西瓜书 5.3

任务详解：

这部分内容，讲述的是神经网络的参数是怎么学习来的，也就是模型的训练过程。这节是正章的核心，毫不夸张的说，是目前整个深度学习的基础和核心，所以这部分的数学推导要求大家。希望大家合上书后，可以完整回忆出整个网络反向传播的流程。如果大家觉得学的不是很懂，建议大家读读《深度学习》花书。

打卡：

(1) 内容：课后习题 5.4。试述式(5.6)中学习率的取值对神经网络训练的影响。

(2) 形式：文字，至少 10 字。

打卡截止日期：7/24

6.3 深度学习初探

学习时长：7/25——7/26

任务简介：学习西瓜书 5.4/5.6

任务详解：

这 5.4 节讲述的是优化问题中的经典问题，即局部极小和全局最小的关系的问题。而这一问题也一直是我们的神经网络经常要考虑的问题，其实优化一个神经网络模型，就是如何求解避免陷入局部极小，找到全局最小点的问题。请大家理解其思维概念就可以，在后续的专业的深度学习过程中，都会接触到的。而 5.6 节就是简单介绍一下深度学习和神经网络的关系，希望大家能够进行简单阅读，了解其关系就可以。在阅读之前，我可以来告诉同学们，理论上讲，或是概念上来说，神经网络只是实现深度学习技术的一种手段，这就意味着还有其他方式可以实现深度学习，但是在如今的工业界和学术界，基本可以认为深度学习就是指的深层的神经网络。大家抓紧学习这个令人兴奋的知识吧。如果大家觉得学的不是很懂，建议大家读读《深度学习》花书。

打卡：

(1) 内容：试述卷积神经网络的基本原理。

(2) 形式：文字，至少 50 字；图片，0-9 张

打卡截止时间：7/26

6.4 了解 sklearn 包中神经网络的使用

学习时长：7/26——7/27

任务名称：实验-sklearn-user guide 1.17.1-1.17.5

任务详解：

对于神经网络来讲，sklearn 给出了简单网络的实现，比如 MLP。虽然 sklearn 不像其他深度学习库那样专业，但是它的使用极其简单，对于新手极其容易上手，所以想学习深度学习的同学，请好好做这节的实验。这部分实验内容是令人兴奋的实验，因为大家做完这节的实验，你们的一只脚就已经踏入了深度学习领域，可能大家的深度学习之旅就从此开始了，所以希望同学们能对布置的章节进行相关阅读，并且认真完成本周的作业。

资料链接：

https://scikit-learn.org/stable/modules/neural_networks_supervised.html

打卡：

(1) 内容：参照链接给的代码或者下图代码，实际动手敲一遍，使用 sklearn 的神经网络接口实现 MNIST 分类，将运行结果截图。

http://scikit-learn.org/stable/auto_examples/neural_networks/plot_mnist_filters.html#sphx-glr-auto-examples-neural-networks-plot-mnist-filters-py

```

import matplotlib.pyplot as plt
from sklearn.datasets import fetch_openml
from sklearn.neural_network import MLPClassifier

print(__doc__)

# Load data from https://www.openml.org/d/554
X, y = fetch_openml('mnist_784', version=1, return_X_y=True)
X = X / 255.

# rescale the data, use the traditional train/test split
X_train, X_test = X[:60000], X[60000:]
y_train, y_test = y[:60000], y[60000:]

# mlp = MLPClassifier(hidden_layer_sizes=(100, 100), max_iter=400, alpha=1e-4,
#                       solver='sgd', verbose=10, tol=1e-4, random_state=1)
mlp = MLPClassifier(hidden_layer_sizes=(50,), max_iter=10, alpha=1e-4,
                    solver='sgd', verbose=10, tol=1e-4, random_state=1,
                    learning_rate_init=.1)

mlp.fit(X_train, y_train)
print("Training set score: %f" % mlp.score(X_train, y_train))
print("Test set score: %f" % mlp.score(X_test, y_test))

fig, axes = plt.subplots(4, 4)
# use global min / max to ensure all weights are shown on the same scale
vmin, vmax = mlp.coefs_[0].min(), mlp.coefs_[0].max()
for coef, ax in zip(mlp.coefs_[0].T, axes.ravel()):
    ax.matshow(coef.reshape(28, 28), cmap=plt.cm.gray, vmin=.5 * vmin,
                vmax=.5 * vmax)
    ax.set_xticks(())
    ax.set_yticks(())

plt.show()

```

(2) 形式：图片，至少 1 张；

截止日期：7/27

6.5 本周学习任务简单总结

学习时长：1 天

任务简介：简单回顾本周学到几个重要知识以及观看作业讲解视频

任务详解：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少。

视频：

作业讲解-神经网络

打卡:

(1) 内容: 请用文字描述, 本周所学知识的重点, 也可以思维导图、手写、电子版截图或者拍照均可, 格式不限

(2) 形式: 文字, 最少 20 字; 图片, 0-9 张;

6.6 参考答案

1. 试述将线性函数 $f(x) = w^t x$ 用作神经元激活函数的缺陷

解答:

神经网络中必须要有非线性的激活函数, 无论是在隐层, 还是输出层, 或者全部都是。如果单用 $f(x) = w^t x$ 作为激活函数, 无论多少层的神经网络会退化成一个线性回归, 只不过是把他复杂化了。

2. 试述使用图 5.2(b)激活函数的神经元与对率回归的联系

解答:

两者都是希望将连续值映射到 $\{0,1\}$ 上, 但由于阶跃函数不光滑, 不连续的性质, 所以才选择了 sigmoid 作为映射函数。不同之处在于激活函数不一定要使用 sigmoid, 只要是非线性的可导函数都可以使用。

3. 试述学习率的取值对神经网络训练的影响

解答:

如果学习率太低, 每次下降的很慢, 使得迭代次数非常多。如果学习率太高, 在后面迭代时会出现震荡现象, 在最小值附近来回波动。

4. 试述卷积神经网络的基本原理

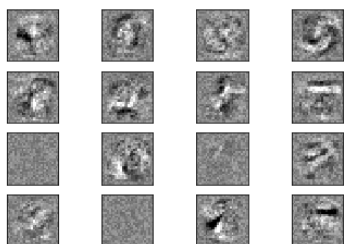
解答: <https://tkhajwjy.github.io/2018/11/24/CNN/>

5. 使用 sklearn 的神经网络接口实现 MNIST 分类, 将运行结果截图

```
Iteration 1, loss = 0.32009978
Iteration 2, loss = 0.15347534
Iteration 3, loss = 0.11544755
Iteration 4, loss = 0.09279764
Iteration 5, loss = 0.07889367
Iteration 6, loss = 0.07170497
Iteration 7, loss = 0.06282111
Iteration 8, loss = 0.05530788
Iteration 9, loss = 0.04960484
Iteration 10, loss = 0.04645355
```

```
D:\MyPythonWork\MachineLearning-WatermelonBook\venv\lib\site-packages\sklearn\network\multilayer_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Training set score: 0.986800
Test set score: 0.970000
```



7 第 7 周

本周学习内容介绍：

本周我们要学习的主要内容为西瓜书的第 2 章模型评估与选择。本章的内容，我并没有给大家安排在学习的开始阶段，因为我深知，这部分内容的理解是建立在你有一定的机器学习实践的经验上的，所以大家在学习过很多模型算法后，我再将本章的内容拿到大家的面前。这章的内容始终都围绕着一个主题展开的，就是如何更加客观的去对一个模型进行评价，也就是模型 A 与模型 B 相比，哪个模型更优秀。本章的内容不涉及数学算法的内容，但是这章的内容在我们的工程实践中是最经常用到的部分，可以说，这章是我们工程实践的一个理论指导，所以想做好工程，请大家认真学习本章。

7.1 经验误差与过拟合

学习时长：7/29——7/30

任务简介：学习西瓜书 2.1

任务详解：

这本小节主要讲述两个概念，一个是模型泛化误差的概念，另一个是模型过拟合的概念。相信大家，在这么久的学习过程中，一定都已经听说了这两个概念，甚至有一部分人已经做了提前了解。我们机器学习一直追求的目标就是它的泛化

能力要强，而不是其对训练集的拟合能力，我们所有的提升模型性能的技术，关注的问题只有一个，就是如何提高模型的泛化能力，请大家时刻记住这个观点。而如果一个模型出现了过拟合的情况，那么这个模型的泛化能力一定是不好的，同时这个概念对于深度学习来讲又是至关重要的，因为深度学习中很多很多技术都是为解决过拟合问题产生的，从而提高模型的泛化能力。所以请大家认真学习这两个概念。

打卡：

（1）内容：过拟合、欠拟合是什么？

（2）形式：文字，至少 10 字；图片，至少 1 张

打卡截止日期：7/30

7.2 评估方法

学习时长：7/30——7/31

任务简介：学习西瓜书 2.2

任务详解：

这部分讲述的主要内容就是：如何客观地划分训练集/验证集/测试集，我们才能训练出一个优秀的模型，让它的泛化能力更加的强。这个内容是我们在做工程问题时，必须面临的问题，我们只有合理的去划分数据集，才能客观的评价一个模型的好坏，才能训练出来一个泛化能力强的模型。所以这章要求大家理解每种采样方法背后的原因。

打卡：

（1）内容：什么是 K 折交叉验证？

（2）形式：文字，至少 20 字或者图片，至少 1 张

打卡截止日期：7/31

7.3 性能度量

学习时长：8/1——8/2

任务名称：学习西瓜书 2.3

任务详解：

这部分内容讲述的内容是评价模型的几种评价指标。我们知道，评价一个事物的好坏，有很多种指标，比如评价一个人，有智商指标/有情商指标/有颜值指标/有品格指标，那么我们应该选哪种指标进行评价更好呢？同理，评价一个机器学习模型也有很多指标，不同的实际问题都会衍生出来不同的指标，但是本节作为基础，给大家介绍几种在我们机器学习领域最为经典的评价指标。大家开始学习吧，看看都有哪些指标！

打卡：

- (1) 内容：解释下精确率、召回率
- (2) 形式：文字，至少 20 字；

7.4 了解 sklearn 包中模型评估方法的使用

学习时长：8/2——8/3

任务名称：实验-sklearn-user guide 3.1.1/3.1.2

任务详解：

这部分的实验，是对前 3 节理论的一个实践。同时，这部分的函数库是我们做机器学习任务经常要用到的，其重要性在于其工程能力比较强，所以要求大家熟练使用这部分的函数。为了增加大家的理解，本周作业也布置了相应的一个实践案例。

资料链接：

http://scikit-learn.org/stable/modules/cross_validation.html

打卡：

- (1) 内容：参照链接给的代码或者下图代码，实际动手敲一遍，将运行结果截图。

http://scikit-learn.org/stable/auto_examples/model_selection/plot_roc_crossval.html#sphx-glr-auto-examples-model-selection-plot-roc-crossval-py

```

print(__doc__)

import numpy as np
from scipy import interp
import matplotlib.pyplot as plt

from sklearn import svm, datasets
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import StratifiedKFold

# #####
# Data IO and generation

# Import some data to play with
iris = datasets.load_iris()
X = iris.data
y = iris.target
X, y = X[y != 2], y[y != 2]
n_samples, n_features = X.shape

# Add noisy features
random_state = np.random.RandomState(0)
X = np.c_[X, random_state.randn(n_samples, 200 * n_features)]

# #####
# Classification and ROC analysis

# Run classifier with cross-validation and plot ROC curves
cv = StratifiedKFold(n_splits=6)
classifier = svm.SVC(kernel='linear', probability=True,
                    random_state=random_state)

tprs = []
aucs = []
mean_fpr = np.linspace(0, 1, 100)

```



```

i = 0
for train, test in cv.split(X, y):
    probas_ = classifier.fit(X[train], y[train]).predict_proba(X[test])
    # Compute ROC curve and area the curve
    fpr, tpr, thresholds = roc_curve(y[test], probas_[ :, 1])
    tprs.append(interp(mean_fpr, fpr, tpr))
    tprs[-1][0] = 0.0
    roc_auc = auc(fpr, tpr)
    aucs.append(roc_auc)
    plt.plot(fpr, tpr, lw=1, alpha=0.3,
             label='ROC fold %d (AUC = %0.2f)' % (i, roc_auc))

    i += 1
plt.plot([0, 1], [0, 1], linestyle='--', lw=2, color='r',
         label='Chance', alpha=.8)

mean_tpr = np.mean(tprs, axis=0)
mean_tpr[-1] = 1.0
mean_auc = auc(mean_fpr, mean_tpr)
std_auc = np.std(aucs)
plt.plot(mean_fpr, mean_tpr, color='b',
         label=r'Mean ROC (AUC = %0.2f  $\pm$  %0.2f)' % (mean_auc, std_auc),
         lw=2, alpha=.8)

std_tpr = np.std(tprs, axis=0)
tprs_upper = np.minimum(mean_tpr + std_tpr, 1)
tprs_lower = np.maximum(mean_tpr - std_tpr, 0)
plt.fill_between(mean_fpr, tprs_lower, tprs_upper, color='grey', alpha=.2,
                 label=r' $\pm$  1 std. dev.')

plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()

```

(2) 形式：图片，至少 1 张；

打卡截止日期：8/3

7.5 本周学习任务简单总结

学习时长：8/4-8/4

任务简介：简单回顾本周学到几个重要知识以及观看作业讲解视频

任务详解：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少。

视频:

作业讲解-模型评估与选择

打卡:

(1) 内容: 请用文字描述, 本周所学知识的重点, 也可以思维导图、手写、电子版截图或者拍照均可, 格式不限

(2) 形式: 文字, 最少 50 字;

打卡截止时间: 8/4

7.6 参考答案

1. 过拟合、欠拟合是什么?

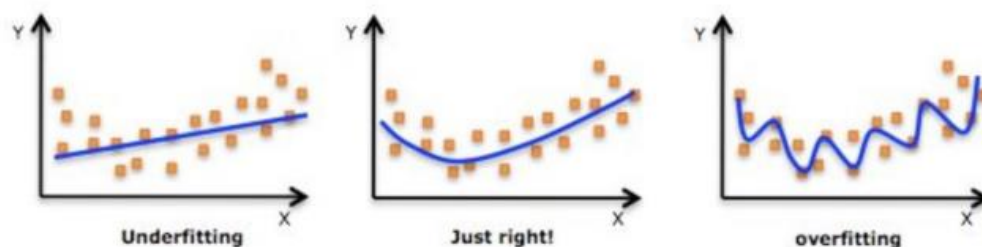
解答:

过拟合 overfitting: 选择的模型包含的参数过多, 以至于该模型对于已知数据预测得很好, 但是对于未知数据预测的很差, 使得训练误差和测试误差之间的差距太大。

- 过拟合的原因是: 将训练样本本身的一些特点当作了所有潜在样本都具有的一般性质, 这会造成泛化能力下降。
- 过拟合无法避免, 只能缓解。因为机器学习的问题通常是 NP 难甚至更难的, 而有效的学习算法必然是在多项式时间内运行完成。如果可以避免过拟合, 这就意味着构造性的证明了 $P=NP$ 。

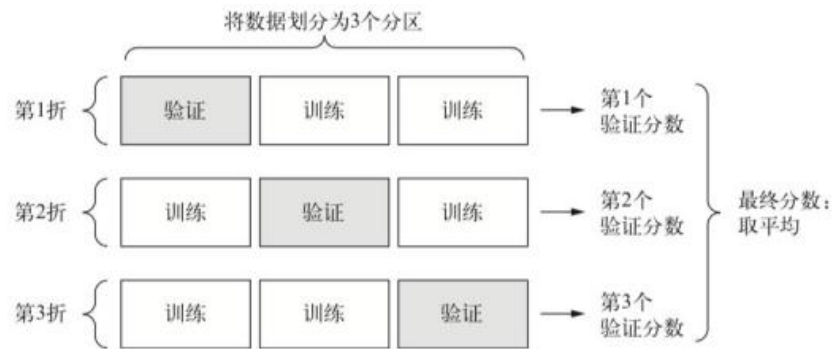
欠拟合 underfitting: 选择的模型包含的参数太少, 以至于该模型对已知数据都预测的很差, 使得训练误差较大。

欠拟合的原因一般是学习能力低下造成的。



2. 什么是 K 折交叉验证?

解答：



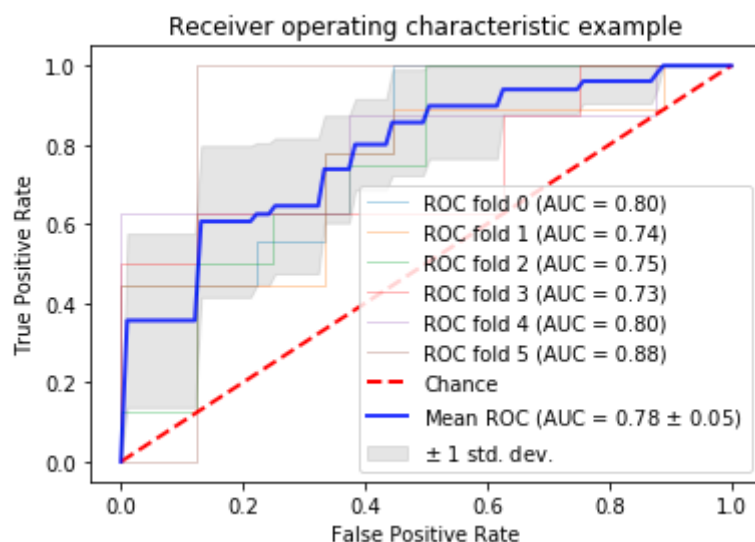
将可用数据划分为 K 个分区(K 通常取 4 或 5)，上图为 3 折交叉验证。实例化 K 个相同的模型，将每个模型在 $K-1$ 个分区上训练，并在剩下的一个分区上进行评估。模型的验证分数等于 K 个验证分数的平均值。

3. 解释下精确率、召回率

解答：

精确率表示的是预测为正的样本中有多少是真正的正样本。那么预测为正就有两种可能了，一种就是把正类预测为正类(TP)，另一种就是把负类预测为正类(FP)，也就是 $P = \frac{TP}{TP+FP}$ 。而召回率表示的是样本中的正例有多少被预测正确了。那也有两种可能，一种是把原来的正类预测成正类(TP)，另一种就是把原来的正类预测为负类(FN)。也就是 $R = \frac{TP}{TP+FN}$ 。

4. 使用 sklearn 的 SVC 接口实现 MNIST 分类模型评估，将运行结果截图



8 第8周

本周学习内容介绍：

本周我们要学习的主要内容为西瓜书的第 10 章特征选择和第 11 章特征降维。我们前边的学习主要是在学习各个机器学习算法，比如逻辑回归/支持向量机/决策树等，可能大家都觉得，这些算法是最重要的，但是我想跟大家说的是，在机器学习工程中，特征工程才是最重要，特征决定着算法的上限，而我们这周要学习的内容，正是特征工程中最为重要和最为基础的两种技术，就是特征选择和特征降维。特征选择和特征降维的目的很简单，就是选择出或变换出更优的特征，从而更利于我们学习算法的学习。而西瓜书在这两部分给出较为全面的理论指导，希望大家认真学习，同时呢，也希望大家能够通过实际项目去学习这部分的内容，明确这两种技术的重要性。

8.1 特征降维

学习时长：8/5——8/6

任务简介：学习西瓜书 10.1/10.2/10.3

任务详解：

特征降维的意思就是原有的一组特征经过数学变换，变换成数量更少的一组特征。通常特征降维主要有两种目的，第一，减少特征维数，使计算开销更小；

第二，使转换后的低维特征更具有特征性，使学习器学习起来更容易。而特征降维主要包括两种方法，有监督降维和无监督降维，在 3.4 节的线性判别分析就是最为典型的监督降维，本节的主成分分析法就是最为典型的无监督降维，请大家对 10.3 节的主成分分析法进行重点的学习。

打卡：

（1）内容：为什么要用 PCA？

（2）形式：文字，至少 50 字；

打卡截止时间：8/6

8.2 特征选择

学习时长：8/8——8/9

任务简介：西瓜书 11.1/11.2/11.3/11.4

任务详解：

这特征选择的目的和特征降维的目的类似，但是特征选择和特征降维具有本质的区别，特征降维的主要特点是通过一个数学变换进行降维，而特征选择就是从众多特征中剔除不重要的特征，从而保留重要的特征。虽然简单粗暴，但是极其容易实现，并易于使用，所以在工程中特征选择的使用频率要高于特征降维。下面给大家说一下特征选择的大致分类，特征选择主要包括 3 种，过滤式选择/包裹式选择/嵌入式选择，大家应该熟记各种方式，并会应用。

打卡：

（1）内容：特征选择包括哪几种？简述一下。

（2）形式：文字，至少 20 字；

打卡截止时间：8/9

8.3 了解 sklearn 包中特征选择和降维算法的使用

学习时长：8/10——8/11

任务简介：实验-sklearn-user guide 1.13.2/1.13.4

任务详解：

这部分的实验，是属于特征选择的内容，其中包含两种方法，一种是过滤式，

另外一种嵌入式选择。这部分的内容在机器学习工程中，会经常使用到，所以大家应该重点掌握。这部分对大家的要求就是，当有新的特征选择问题摆在你面前时，你应该有代码能力进行特征选择，其中的案例部分大家根据个人能力及时间进行选择学习即可，不做具体要求。

资料链接：

http://scikit-learn.org/stable/modules/feature_selection.html

打卡：

（1）内容：参照链接给的代码或者下图代码，实际动手敲一遍，将运行结果截图。

https://scikit-learn.org/stable/auto_examples/feature_selection/plot_feature_selection.html#sphx-glr-auto-examples-feature-selection-plot-feature-selection-py

```
print(__doc__)

import numpy as np
import matplotlib.pyplot as plt

from sklearn import datasets, svm
from sklearn.feature_selection import SelectPercentile, f_classif

# #####
# Import some data to play with

# The iris dataset
iris = datasets.load_iris()

# Some noisy data not correlated
E = np.random.uniform(0, 0.1, size=(len(iris.data), 20))

# Add the noisy data to the informative features
X = np.hstack((iris.data, E))
y = iris.target

plt.figure(1)
plt.clf()

X_indices = np.arange(X.shape[-1])
```

```

# #####
# Univariate feature selection with F-test for feature scoring
# We use the default selection function: the 10% most significant features
selector = SelectPercentile(f_classif, percentile=10)
selector.fit(X, y)
scores = -np.log10(selector.pvalues_)
scores /= scores.max()
plt.bar(X_indices - .45, scores, width=.2,
        label=r'Univariate score ($-Log(p_{value}))$', color='darkorange',
        edgecolor='black')

# #####
# Compare to the weights of an SVM
clf = svm.SVC(kernel='linear')
clf.fit(X, y)

svm_weights = (clf.coef_ ** 2).sum(axis=0)
svm_weights /= svm_weights.max()

plt.bar(X_indices - .25, svm_weights, width=.2, label='SVM weight',
        color='navy', edgecolor='black')

clf_selected = svm.SVC(kernel='linear')
clf_selected.fit(selector.transform(X), y)

svm_weights_selected = (clf_selected.coef_ ** 2).sum(axis=0)
svm_weights_selected /= svm_weights_selected.max()

plt.bar(X_indices[selector.get_support()] - .05, svm_weights_selected,
        width=.2, label='SVM weights after selection', color='c',
        edgecolor='black')

plt.title("Comparing feature selection")
plt.xlabel('Feature number')
plt.yticks(())
plt.axis('tight')
plt.legend(loc='upper right')
plt.show()

```

(2) 形式：图片，至少 1 张；

打卡截止时间：8/11

8.4 本周学习任务简单总结

学习时长：8/11-8/11

任务简介：简单回顾本周学到几个重要知识以及观看作业讲解视频

任务详解：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时

做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少

视频：

作业讲解-特征选择

打卡：

（1）内容：请用文字描述，本周所学知识的重点，也可以思维导图、手写、电子版截图或者拍照均可，格式不限

（2）形式：文字，最少 50 字；

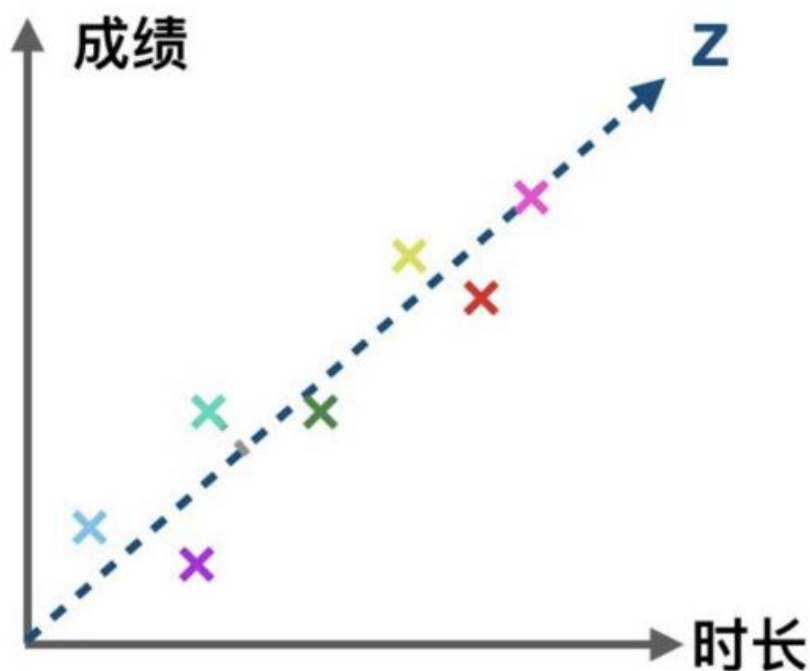
打卡截止时间：8/11

8.5 参考答案

1. 为什么要用 PCA?

解答：

PCA 是为了降低特征的维度。在实际工作中，可能会遇到特征维数过多，且部分特征具有一定相关性。如：在一个学生成绩数据集中，一个特征是学习时长，另一个特征是成绩，一般来讲，学习时间越长，越容易取得好成绩，即时长与成绩呈正相关。因此没必要用两个维度（时长和成绩）表达这一特征。我们可以找出另一个维度，表现这两个特征，且不会带来过多信息损失。

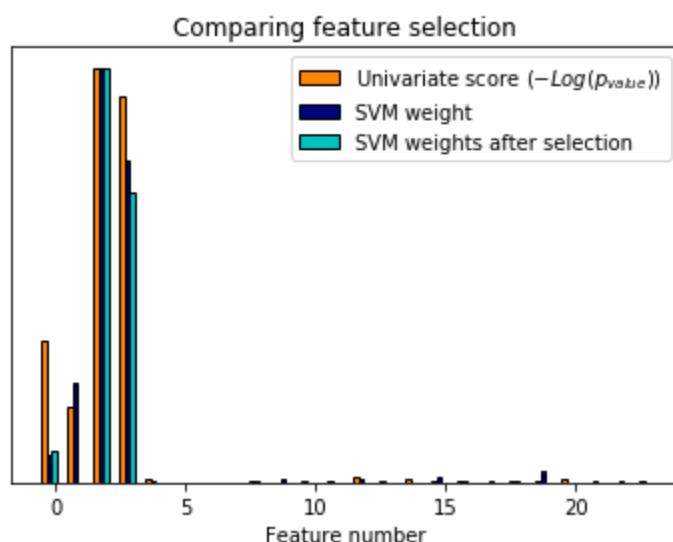


由图可知，可以用数据点在 z 轴上的投影表达时长与成绩两个维度的数据。一个平面上的数据就被映射到了一条线当中，即二维特征向一维特征的转换。

2. 特征选择包括哪几种？简述一下

- 过滤式选择：选择与目标变量相关性较强的特征。缺点:没有考虑到特征之间的关联作用。
- 包裹式选择：把特征选择看做一个特征子集搜索问题，筛选各种特征子集，用模型评估效果。
- 嵌入式选择：根据模型来分析特征的重要性(有别于上面的方式，是从生产上的模型权重等)。最常见的方式为用正则化方式来做特征选择。

3. 使用 sklearn 的 SVM 接口实现 IRIS 数据集特征选择和降维算法的应用，将运行结果截图



9 第9周

本周学习内容介绍：

本周我们要学习的主要内容为西瓜书的第8章集成学习。这个章节是全书最为重要章节之一，在比赛中用集成学习的方法去提高模型的性能是必备的一项技术，在工程中，当实际问题对计算实时性要求不高的场合，集成学习也被广泛应用于生产中用于提升模型的泛化能力。可以说本章的所有内容都是重要内容，请大家认真对待学习。

9.1 集成学习

学习时长：8/12——8/14

任务简介：西瓜书 8.1/8.2/8.3

任务详解：

本小节学习内容主要包括两大部分，一部分是 **boosting**，另外一部分是 **bagging** 与随机森林，这两部分的集成思想是有区别的，请大家重点关注一下这两种集成方法的主要区别是什么？其中，我们所熟知的 **XGBoost** 和 **lightGBM** 就是基于 **boosting** 技术进行构建的，准确来说是基于 **GBDT** 算法构建的，而西瓜书中对于 **boosting** 的介绍，不是很详细，这里我强烈建议大家去看一下李航老师的统计学习方法的第8章，值得我们去认真阅读推敲。

打卡：

(1) 内容：集成学习分为哪几种？请简述它们的思想（作业答案以及讲解视频会在周日公布）

(2) 形式：文字，最少 50 字；图片，0-9 张。

打卡截止时间：8/14

9.2 结合策略

学习时长：8/15——8/16

任务简介：西瓜书 8.4/8.5

任务详解：

这部分讲的内容就是想告诉我们怎么才能实现多个好而不同的单模型，在得到多个好而不同的模型后，我们应该使用什么样的融合策略才能得到一个最优的集成模型，这就分别对应着 8.5 节和 8.4 节的内容。其中 8.5.1 节，是对“好而不同”的理论分析，可以说很是抽象，所以大家简单阅读这部分内容即可，不做深入要求。

打卡：

(1) 内容：简述 Stacking 算法

(2) 形式：文字，最少 10 字

打卡截止时间：8/16

9.3 实验-lightGBM 的使用

学习时长：8/16——8/18

任务详解：

这部分的实验，重点给大家引入一个最优秀的集成学习的工具，lightGBM，这个工具无论在比赛中还是在生产环境中，都是一把利器，能够很好的解决很多机器学习问题，这都得益于其优异的性能，可以说这是机器学习者必须所掌握的。在这部分，大家要阅读链接的内容，并完成今日的作业。

资料链接：

<https://lightgbm.apachecn.org/#/>

打卡:

(1) 内容: 参照链接给的代码或者下图代码, 实际动手敲一遍, 将运行结果截图。

https://github.com/Microsoft/LightGBM/blob/master/examples/python-guide/simple_example.py

```
import lightgbm as lgb
import pandas as pd
from sklearn.metrics import mean_squared_error

print('Loading data...')
# load or create your dataset
df_train = pd.read_csv('data/regression.train', header=None, sep='\t')
df_test = pd.read_csv('data/regression.test', header=None, sep='\t')

y_train = df_train[0]
y_test = df_test[0]
X_train = df_train.drop(0, axis=1)
X_test = df_test.drop(0, axis=1)

# create dataset for lightgbm
lgb_train = lgb.Dataset(X_train, y_train)
lgb_eval = lgb.Dataset(X_test, y_test, reference=lgb_train)
# specify your configurations as a dict
params = {
    'boosting_type': 'gbdt',
    'objective': 'regression',
    'metric': {'l2', 'l1'},
    'num_leaves': 31,
    'learning_rate': 0.05,
    'feature_fraction': 0.9,
    'bagging_fraction': 0.8,
    'bagging_freq': 5,
    'verbose': 0
}
print('Starting training...')
# train
```

```

gbm = lgb.train(params,
                lgb_train,
                num_boost_round=20,
                valid_sets=lgb_eval,
                early_stopping_rounds=5)

print('Saving model...')
# save model to file
gbm.save_model('model.txt')

print('Starting predicting...')
# predict
y_pred = gbm.predict(X_test, num_iteration=gbm.best_iteration)
# eval
print('The rmse of prediction is:', mean_squared_error(y_test, y_pred) ** 0.5)

```

(2) 形式：图片，至少 1 张。

打卡截止时间：8/18

9.4 本周学习任务简单总结

学习时长：8/18——8/18

任务简介：简单回顾本周学到几个重要知识以及观看作业讲解视频

任务详解：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少

打卡：

(1) 内容：请用文字描述，本周所学知识的重点，也可以思维导图、手写、电子版截图或者拍照均可，格式不限

(2) 形式：文字，最少 50 字；

打卡截止时间：8/18

9.5 参考答案

1. 集成学习分为哪几种？请简述它们的思想

● Boosting

Boosting 方法训练基分类器采用串行方法，各分类器之间有依赖。它的训练思想是迭代式的学习，类似人类学习一样。基本思路是将基本分类器层层叠加，每一

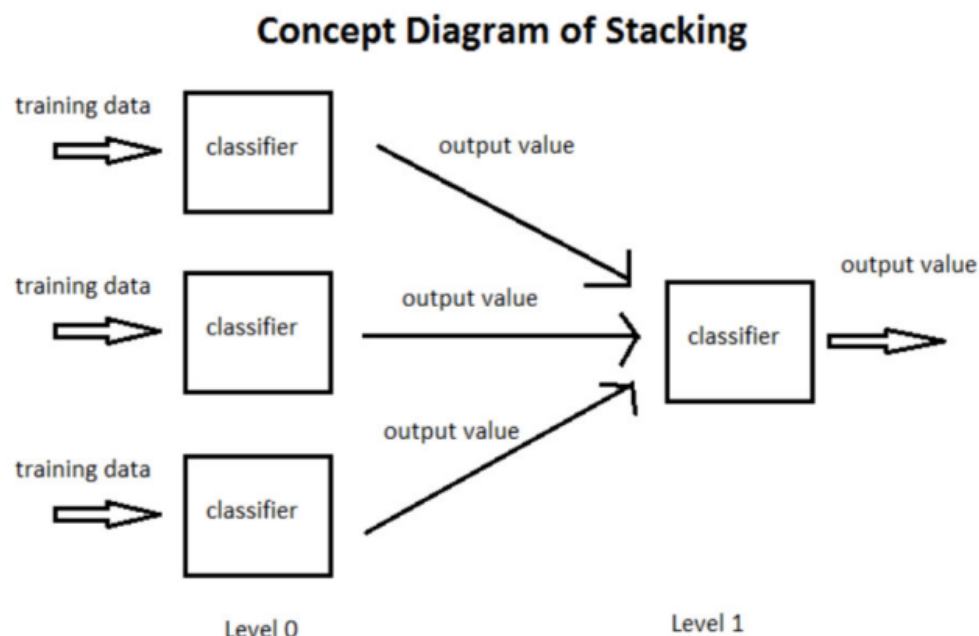
层在训练的时候，对前一层的基分类器分错的样本给予更高的权重。测试时，根据各层分类器的结果的加权得到最终结果。

- Bagging

Bagging 方法训练基分类器采用并行方法，各分类器之间无强依赖。它的训练思想是集体投票决策。基本思路是将训练集分为若干子集，每个基分类器分别进行学习，在最后通过投票方式做最终的决策。

2. 简述 Stacking 算法

将多种 predictor 结果作为特征训练



$$\hat{y}_1 = f_1(x_1, x_2, \dots)$$

$$\hat{y}_2 = f_2(x_1, x_2, \dots)$$

...



$$\hat{y}_e = \text{sign}(\sum \alpha_i \hat{y}_i)$$

$f_e() = \text{majority}$ — 等价于vote(majority vote)

$f_e() = \text{linear}$ — 等价于加权平均

3. 使用 lightGBM，将运行结果截图

```
Loading data...
Starting training...
[1]      valid_0's 12: 0.24288      valid_0's 11: 0.491812
Training until validation scores don't improve for 5 rounds.
[2]      valid_0's 12: 0.239307      valid_0's 11: 0.48798
[3]      valid_0's 12: 0.235559      valid_0's 11: 0.483905
[4]      valid_0's 12: 0.230771      valid_0's 11: 0.478533
[5]      valid_0's 12: 0.226297      valid_0's 11: 0.47333
[6]      valid_0's 12: 0.223692      valid_0's 11: 0.470079
[7]      valid_0's 12: 0.220941      valid_0's 11: 0.466768
[8]      valid_0's 12: 0.217982      valid_0's 11: 0.462971
[9]      valid_0's 12: 0.215351      valid_0's 11: 0.459474
[10]     valid_0's 12: 0.213064      valid_0's 11: 0.45619
[11]     valid_0's 12: 0.211053      valid_0's 11: 0.4532
[12]     valid_0's 12: 0.209336      valid_0's 11: 0.450526
[13]     valid_0's 12: 0.207492      valid_0's 11: 0.447621
[14]     valid_0's 12: 0.206016      valid_0's 11: 0.445178
[15]     valid_0's 12: 0.204677      valid_0's 11: 0.442718
[16]     valid_0's 12: 0.203224      valid_0's 11: 0.440293
[17]     valid_0's 12: 0.201186      valid_0's 11: 0.437015
[18]     valid_0's 12: 0.199626      valid_0's 11: 0.434725
[19]     valid_0's 12: 0.198689      valid_0's 11: 0.432653
[20]     valid_0's 12: 0.197377      valid_0's 11: 0.430279
Did not meet early stopping. Best iteration is:
[20]     valid_0's 12: 0.197377      valid_0's 11: 0.430279
Saving model...
Starting predicting...
The rmse of prediction is: 0.44427172467679765
```

10 第 10 周

本周学习内容介绍：

本周我们要学习的知识要和之前的知识做一个分割线了，为什么要做分割线呢？因为，第 9 章的聚类，属于无监督问题，是严格区分前边的有监督问题的。而第 14 章的内容属于序列到序列的问题模型，不再是单一标注问题，但仍然属于监督学习的范畴。我们只是拿出来这两章比较简单的章节进行学习，希望大家能够区分本周的内容和之前学习的内容分别适合解决什么样的问题。

10.1 聚类

学习时长：8/19——8/20

任务简介：西瓜书 9.1/9.2/9.3/9.4.1

任务详解：

本小节学习内容是聚类算法，聚类算法主要包括原型聚类/密度聚类/层次聚类，在西瓜书中都应相应的介绍，在 **SKLearn** 库中也都有相应的代码实现。但是本次学习只要求大家掌握最简单的原型聚类算法中的 **k** 均值聚类算法，希望大家可以了解聚类的思想。若以后的工程问题中，同学们遇到聚类问题，希望大家可以回来认真看看其他稍微复杂的聚类算法。

视频：

GMM

打卡：

(1) 内容：简述 **K** 均值算法步骤

(2) 形式：图片，至少 1 张。

打卡截止时间：8/20

10.2 HMM

学习时长：8/22——8/23

任务简介：西瓜书 14.1

任务详解：

这部分讲的算法主要是针对序列标注的问题，而隐马尔科夫模型在自然语言处理/语音识别方面都有重大的应用，同时它还适合解决跟时序相关的问题，希望大家能够认真学习其算法原理，同时对于隐马尔科夫的数学推导，统计学习方法中的第 10 章的推导更为详细，大家可以进行重点学习，对于这部分的实战，我会给大家布置一个作业，还请大家完成。而本章的后续章节，我们训练营不做要求，同学们根据个人自身情况进行选择学习。

视频：

hmm-1

hmm-2

hmm-3

打卡：

(1) 内容: HMM 解决哪三大问题?

(2) 形式: 文字, 至少 50 字;

打卡截止时间: 8/23

10.3 了解 sklearn 包中 K-means 算法的使用

学习时长: 8/24——8/25

任务详解:

这部分的实验, 主要是针对 K-means 算法进行实践, 大家要先对资料链接进行阅读, 然后再将打卡中的代码进行阅读/编辑, 并运行。

资料链接:

<https://sklearn.apachecn.org/docs/0.21.3/22.html>

打卡:

(1) 内容: 参照链接给的代码或者下图代码, 实际动手敲一遍, 将运行结果截图。

http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_assumptions.html#sphx-glr-auto-examples-cluster-plot-kmeans-assumptions-py

```

print(__doc__)

# Author: Phil Roth <mr.phil.roth@gmail.com>
# License: BSD 3 clause

import numpy as np
import matplotlib.pyplot as plt

from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

plt.figure(figsize=(12, 12))

n_samples = 1500
random_state = 170
X, y = make_blobs(n_samples=n_samples, random_state=random_state)

# Incorrect number of clusters
y_pred = KMeans(n_clusters=2, random_state=random_state).fit_predict(X)

plt.subplot(221)
plt.scatter(X[:, 0], X[:, 1], c=y_pred)
plt.title("Incorrect Number of Blobs")

# Anisotropically distributed data
transformation = [[0.60834549, -0.63667341], [-0.40887718, 0.85253229]]
X_aniso = np.dot(X, transformation)
y_pred = KMeans(n_clusters=3, random_state=random_state).fit_predict(X_aniso)

```

```

plt.subplot(222)
plt.scatter(X_aniso[:, 0], X_aniso[:, 1], c=y_pred)
plt.title("Anisotropically Distributed Blobs")

# Different variance
X_varied, y_varied = make_blobs(n_samples=n_samples,
                                cluster_std=[1.0, 2.5, 0.5],
                                random_state=random_state)
y_pred = KMeans(n_clusters=3, random_state=random_state).fit_predict(X_varied)

plt.subplot(223)
plt.scatter(X_varied[:, 0], X_varied[:, 1], c=y_pred)
plt.title("Unequal Variance")

# Unevenly sized blobs
X_filtered = np.vstack((X[y == 0][:500], X[y == 1][:100], X[y == 2][:10]))
y_pred = KMeans(n_clusters=3,
                random_state=random_state).fit_predict(X_filtered)

plt.subplot(224)
plt.scatter(X_filtered[:, 0], X_filtered[:, 1], c=y_pred)
plt.title("Unevenly Sized Blobs")

plt.show()

```

(2) 形式：图片，至少 1 张

打卡截止时间：8/25

10.4本周学习任务简单总结

学习时长：1 天

任务简介：简单回顾本周学到几个重要知识以及观看作业讲解视频

任务详解：

每一周的学习任务都比较重，第一次学过之后特别容易忘，所以在周日及时做一个要点回顾，会让学习效率大大的提升，不会的知识也会越来越少

打卡：

(1) 内容：请用文字描述，本周所学知识的重点，也可以思维导图、手写、电子版截图或者拍照均可，格式不限

(2) 形式：文字，最少 50 字；

打卡截止时间：8/25

10.5参考答案

1. 简述 K 均值算法步骤

解答：

K 均值聚类的核心目标是将给定的数据集划分成 K 个簇，并给出每个数据对应的簇中心点，算法的具体步骤描述如下：

(1) 数据预处理，如归一化、离群点处理等。

(2) 随机选取 K 个簇中心，记为 $\mu_1^{(0)}, \mu_2^{(0)}, \dots, \mu_k^{(0)}$ 。

(3) 定义代价函数： $J(c, \mu) = \min_{\mu} \min_c \sum_{i=1}^M \|x_i - \mu_{c_i}\|^2$ 。

(4) 令 $t = 0, 1, 2, \dots$ 为迭代步数，重复下面过程直到 J 收敛：

- 对于每一个样本 x_i ，将其分配到距离最近的簇

$$c_i^{(t)} \leftarrow \arg \min_k \|x_i - \mu_k^{(t)}\|^2$$

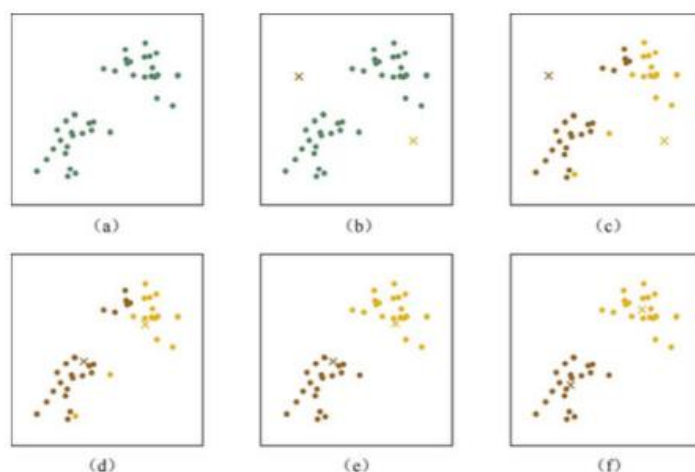
- 对于每一个类簇 k，重新计算该类簇的中心

$$\mu_k^{(t+1)} \leftarrow \arg \min_{\mu} \sum_{i: c_i^{(t)} = k} \|x_i - \mu\|^2$$

K 均值算法在迭代时，假设当前 J 没有达到最小值，那么首先固定簇中心 $\{\mu_k\}$ ，

调整每个样例 x_i 所属的类别 c_i 来让J函数减少；然后固定 $\{c_i\}$ ，调整簇中心 $\{\mu_k\}$ 使J减小。这两个过程交替循环，J单调递减；当J递减到最小值时， $\{\mu_k\}$ 和 $\{c_i\}$ 也同时收敛。

下图是 K-means 算法的一个迭代过程示意图。首先，给定二维空间上的一些样本点（图(a)），直观上这些点可以被分成两类；接下来，初始化两个中心点（图(b)的棕色和黄色叉子代表中心点），并根据中心点的位置计算每个样本所属的簇（图(c)用不同颜色表示）；然后根据每个簇中的所有点的平均值计算新的中心点位置（图(d)）；图(e)和图(f)展示了新一轮的迭代结果：在经过两轮的迭代之后，算法基本收敛。



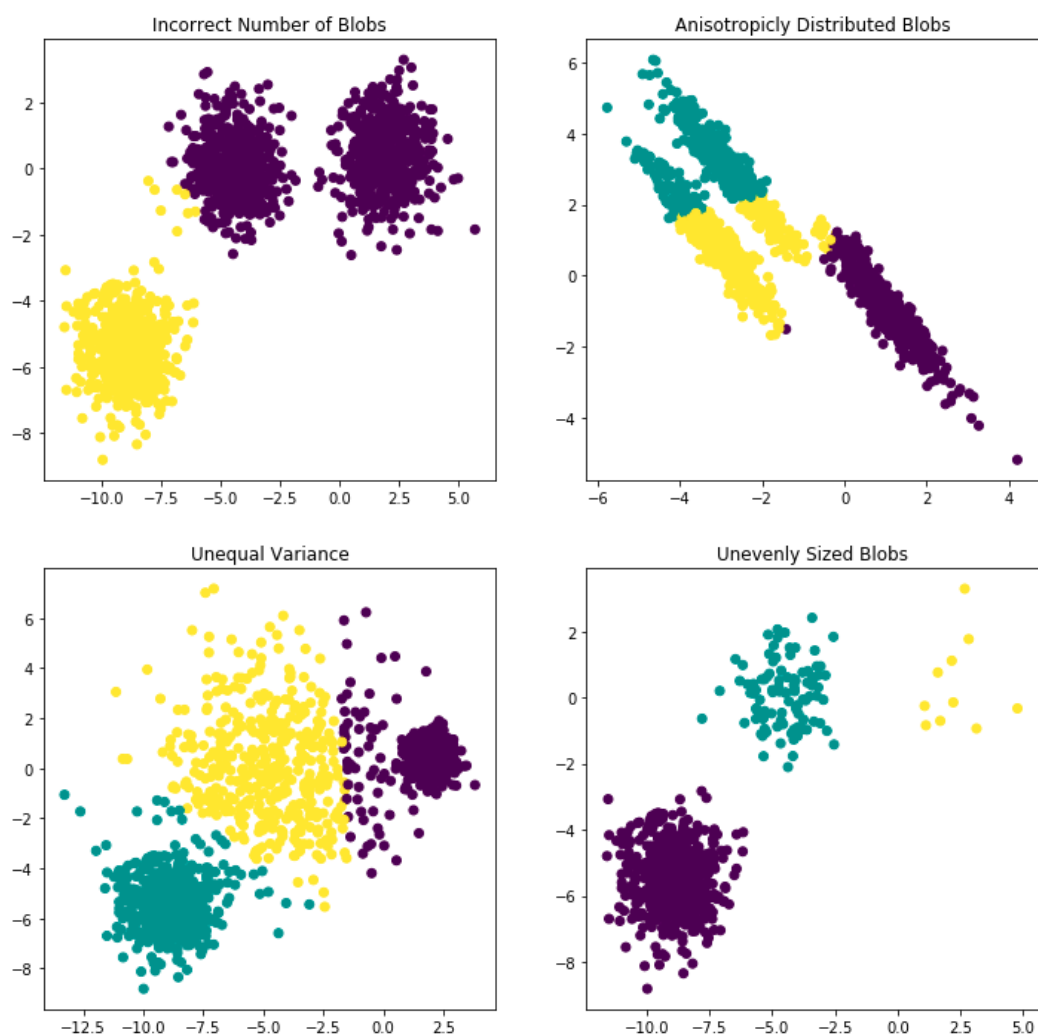
2. HMM 解决哪三大问题？

解答：

隐马尔可夫模型包括概率计算问题、预测问题、学习问题三个基本问题：

- (1) 概率计算问题：已知模型的所有参数，计算观测序列 Y 出现的概率，可使用前向和后向算法求解。
- (2) 预测问题：已知模型所有参数和观测序列 Y ，计算最可能的隐状态序列 X ，可使用经典的动态规划算法——维特比算法类求解最可能的状态序列。
- (3) 学习问题：已知观测序列 Y ，求解使得该观测序列概率最大的模型参数，包括隐状态序列、隐状态之间的转移概率分布以及从隐状态到观测状态的概率分布，可使用 Baum-Welch 算法进行参数的学习，Baum-Welch 算法是最大期望算法的一个特例。

3. 使用 sklearn 的 K-means 接口实现聚类，将运行结果截图



11 第 11 周

本周学习内容介绍：

各位同学们好，本周任务为我们西瓜书的最后一次任务，本周的任务内容是强化学习。在完成本章的任务后，还会有一部分自由时间任同学们自己安排，同学们可以选择对学过的知识进行复习，也可以对之前未学习的章节进行学习，或者是对本章的内容进行更深入的学习。

整个西瓜书，计算学习/半监督学习/规则学习这三章的内容，我们训练营没有涉及，因为这三章的内容相比较而言，不是那么重要，甚至可以说在工程中，很少有人会涉及到这部分的技术。而本章的内容，强化学习，不同与之前 9 周学过的算法，本质的讲，学习策略不同，同时这部分的内容也是更加难懂，更加的

前沿，所以这部分的学习对大家的要求自然而言是了解性的，不期望同学们在一周或是两周的时间内搞懂强化学习，只是希望同学们能够对强化学习有一定的了解。当然了，有很多同学对强化学习特别的偏爱，对于这部分同学，我希望你们能够深入的研究下去，因为强化学习是一个令人兴奋的技术，特别是对于机器人技术的发展。

本章，16.1/16.3/16.4 是重点内容，希望大家可以认真阅读，此外本次作业给大家引入了一个强化学习的宝藏，给各位探索家开启宝藏之门！

11.1 任务与奖赏

学习时长：1 天

任务简介：西瓜书 16.1

任务详解：本节内容主要告诉大家什么是强化学习，特此作者举了一个例子，学习如何种西瓜的例子，即学习种西瓜的过程就是一个典型的强化学习的案例，希望大家可以细细品味。

打卡：

（1）内容：强化学习与有监督学习差别

（2）形式：文字，最少 20 字；

打卡截止时间：8/27

11.2 K-摇臂赌博机和天池 o2o 比赛初级

学习时长：8/27——8/28

任务简介：学习西瓜书 16.2，搭建 Python 开发环境，学习天池 o2o 优惠券使用预测比赛初级源代码，运行程序，提交结果，查看成绩。

任务详解：

（1）本节内容主要是讲述了一个最为简单的强化学习的案例：K-摇臂赌博机，为什么说这个是最简单的强化学习案例呢？因为一般的强化学习过程，智能体（机器人）要连续完成 n 个动作，才是一个完整的过程，比如种西瓜，要浇水、施肥、除草、杀虫等 n 个动作。而本节例子，要完成的动作只有一个，就是选择 K 个摇臂中的一个。

(2) 注册天池账号，报名参加“天池新人实战赛 o2o 优惠券使用预测”比赛。
提交给定的结果样例，查看成绩。目的是让大家走一遍比赛流程。

搭建 Python 开发环境 (Python 开发环境配置教程)

<https://shimo.im/docs/W5pX5mENS20DCquh>

Jupyter Notebook 速成手册

上: <https://mp.weixin.qq.com/s/O2nTGOtqGR-V33-YJgPgJQ>

下: <https://mp.weixin.qq.com/s/AwSzkjlpwvdUzh6CmHq6AQ>

天池 o2o 优惠券使用预测比赛初级源代码和数据集

链接: <https://pan.baidu.com/s/1JkMCOmcmXIaOUoC9L6c3Vg>

提取码: [hhen](#)

视频:

天池 o2o 优惠券使用预测比赛解析 (初级) 视频讲解

天池 o2o 优惠券使用比赛预测分析比赛

打卡:

(1) 内容: (16.1 题目)

用于 K-摇臂赌博机的 UCB (Upper Confidence Bound, 上置信界) 方法每次选择 $Q(k) + UC(k)$ 最大的摇臂, 其中 $Q(k)$ 为摇臂 k 当前的平均奖赏, $UC(k)$ 为置信区间, 例如:

$$Q(k) + \sqrt{\frac{2 \ln n}{n_k}}$$

其中 n 为已执行所有摇臂的总次数, n_k 为已执行摇臂 k 的次数, 试比较 UCB 方法与 ϵ -贪心法和 Softmax 方法的异同。

形式: 文字, 最少 20 字;

(2) 内容: 天池 o2o 优惠券使用预测比赛初级源代码, 运行程序, 提交结果, 查看成绩。

形式: 可以是图片+文字, 或者思维导图。

打卡截止时间: 8/28

11.3 有/无模型学习和天池 o2o 比赛进阶

任务时长：8/29——8/30

任务简介：阅读西瓜书 16.3/16.4，搭建 Python 开发环境，学习天池 o2o 优惠券使用预测比赛进阶源代码，运行程序，提交结果，查看成绩。

任务详解：

（1）这 2 小节内容主要讲述有模型学习和无模型学习，希望大家在概念上能够理解这两节内容的区别，对于这部分的学习，希望大家能够大致了解就可以。

（2）搭建 Python 开发环境，学习天池 o2o 优惠券使用预测比赛进阶源代码，运行程序，提交结果，查看成绩。

天池 o2o 优惠券使用预测比赛进阶源代码和数据集下载：

链接：<https://pan.baidu.com/s/1dqIDpyRatBdKkOJSTkvdpw>

提取码：htxv

视频：

天池 o2o 优惠券使用预测比赛解析（进阶）视频讲解

打卡：

（1）内容：（16.4 题目）

在没有 MDP 模型时，可以先学习 MDP 模型（例如使用随机策略进行采样，从样本中估计出转移函数和奖赏函数），然后再使用有模型强化学习方法。试述该方法与无模型强化学习方法的优缺点。

形式：文字，最少 20 字；

（2）内容：搭建 Python 开发环境，学习天池 o2o 优惠券使用预测比赛进阶源代码，运行程序，提交结果，查看成绩。

形式：可以是图片+文字，或者思维导图。

打卡截止时间：8/30

11.4 本周学习任务简单总结

学习时长：9/1——9/1

任务简介：简单回顾本周学到几个重要知识

任务详解：每学习就要结束了，恭喜大家通关本次西瓜书。

打卡：

(1) 内容：请用文字描述，本周所学知识的重点，也可以思维导图、手写、电子版截图或者拍照均可，格式不限

(2) 形式：文字，最少 50 字；

打卡截止时间：9/1

11.5 参考答案

1. 强化学习与有监督学习差别

解答：

强化学习中没有有标记的样本，没有人告诉机器在什么状态做什么动作，只有在最终结果揭晓之后，才能反思之前动作是否正确来进行学习，因此，强化学习在某种意义上可以看成“延迟标记信息”的监督学习问题。

2. 题 16.1

用于 k -摇臂赌博机的 UCB (Upper Confidence Bound, 上置信界) 方法每次选择 $Q(k) + UC(k)$ 最大的摇臂，其中 $Q(k)$ 为摇臂 k 当前的平均奖赏， $UC(k)$ 为置信区间，例如：

$$Q(k) + \sqrt{\frac{2 \ln n}{n_k}}$$

其中 n 为已执行所有摇臂的总次数， n_k 为已执行摇臂 k 的次数，试比较 UCB 方法与 ϵ -贪心法和 Softmax 方法的异同。

解答：

相同点：都能实现探索和利用的折中选取。

不同点：

- UCB 方法并没有参数，不需要对参数进行初始估计和调整。
- UCB 方法中的 $UC(k)$ 一项， n_k 初始值为 0，作为分母会导致每个摇臂的值为无穷大。那么可以看出 UCB 方法会先将每一个摇臂都尝试一次，也就相当于一次“全探索”。

- UCB 方法中摇臂选取全部由前面选择的摇臂的奖赏值决定，而不是概率。

3. 题 16.4

在没有 MDP 模型时，可以先学习 MDP 模型（例如使用随机策略进行采样，从样本中估计出转移函数和奖赏函数），然后再使用有模型强化学习方法。试述该方法与免模型强化学习方法的优缺点。

解答：

若没有 MDP 模型，根据采样的方式学习出它的 P 和 R ，但是有限的样本无法得出精确的 PR 值，而且不同的采样也会有不同的值，导致最终强化学习的结果不同。蒙特卡罗强化方法考虑采样轨迹，但每次都要采样一个轨迹才能更新策略的估计值，效率很低。