计算机与信息工程学院实验报告（一）

姓名：王赫 学号：1828070097 专业：数据科学与大数据技术(明德计划) 年级：2018级

课程：机器学习与数据挖掘 主讲教师：罗慧敏 辅导教师：\_\_\_\_\_\_\_

实验题目： 作业5

实验目的： 集成学习：Hoeffding不等式推导应用

聚类学习：K-Means算法应用

实验内容：

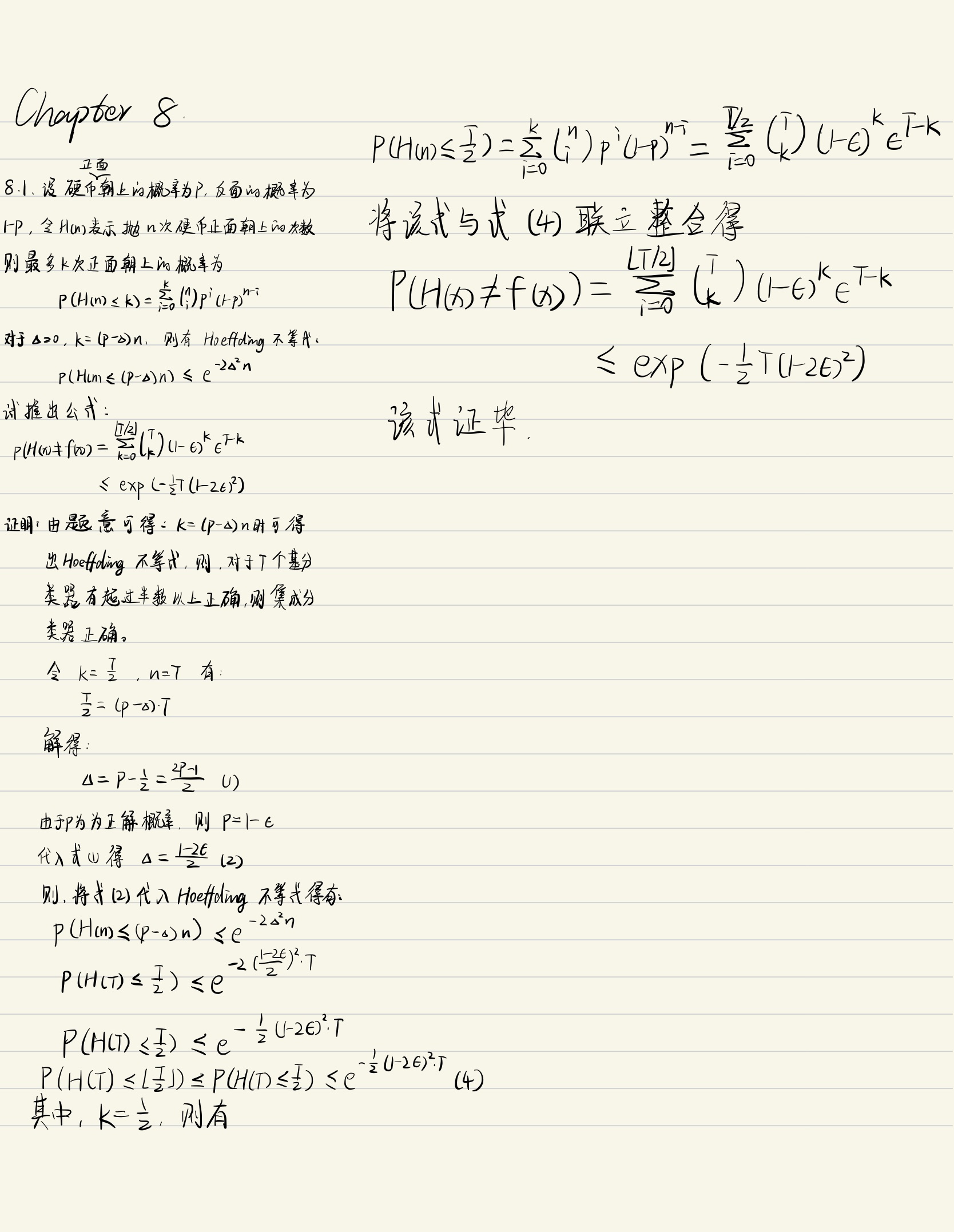
作业内容：

作业题8.1

K-Means对西瓜数据集4.0进行聚类划分（5个）

实验数据记录：

作业题8.1



K-Means对西瓜数据集4.0进行聚类划分（5个）

# 1.导入依赖包

import numpy as np

import matplotlib.pyplot as plt

import matplotlib as mpl

import scipy.io

# 2.加载数据

def loaddata():

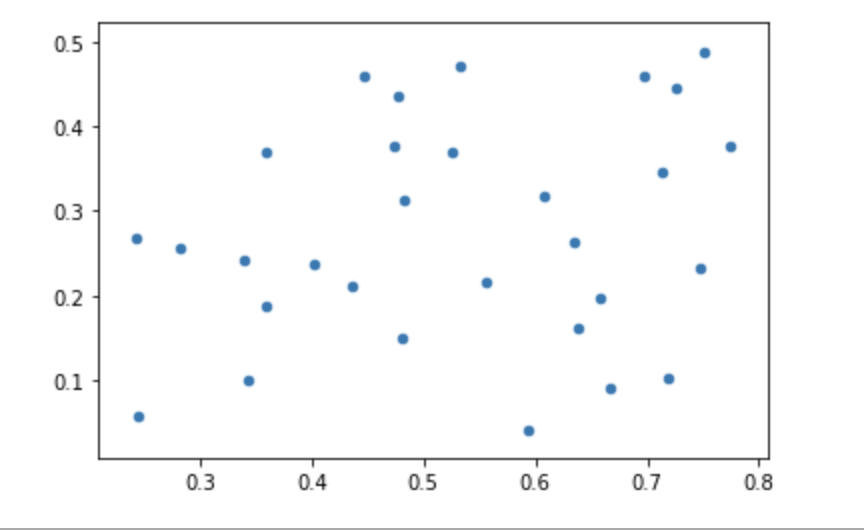
data = np.loadtxt('watermelon\_4.txt',delimiter=',')

return data

X = loaddata()

plt.scatter(X[:, 0], X[:, 1], s=20)

plt.show()



# 3.随机初始化中心点

def kMeansInitCentroids(X, k):

#从X的数据中随机取k个作为中心点

index = np.random.randint(0, len(X)-1, k)

return X[index]

# 4.计算数据点到中心点的距离，并判断该数据点属于哪个中心点--找到距离随即中心点距离最小的数据点，即为本次集群划分的中心点

def findClosestCentroids(X, centroids):

#idx中数据表明对应X的数据是属于哪一个中心点的

idx = np.zeros(len(X)).reshape(X.shape[0],-1)

for i in range(len(X)):

minDistance = float('inf') # 初始为无限大

index = 0

for k in range(len(centroids)):

# 遍历求解欧几里得距离

dis = np.sum(np.power(X[i]-centroids[k], 2))

if dis < minDistance:

minDistance = dis

index = k

idx[i] = index

return idx

# 5.重新计算中心点位置

def computeCentroids(X, idx):

k = set(np.ravel(idx).tolist()) #找到所有聚类中心索引

k = list(k)

centroids = np.ndarray((len(k),X.shape[1]))

for i in range(len(k)):

#选择数据X中类别为k[i]的数据

data = X[np.where(idx==k[i])[0]]

#重新计算聚类中心

centroids[i] = np.sum(data,axis=0)/len(data)

return centroids

# 6.K-Means算法实现

def k\_means(X, k, max\_iters):

initial\_centroids = kMeansInitCentroids(X,k)

for i in range(max\_iters):

if i == 0:

centroids = initial\_centroids

#计算样本到簇群中心的距离，并返回每个样本所属的簇中心

idx = findClosestCentroids(X, centroids)

#更新数据中心点

centroids = computeCentroids(X, idx)

return idx,centroids

idx,centroids = k\_means(X, 5, 500)

print(idx)

print(centroids)



cm\_dark = mpl.colors.ListedColormap(['g', 'r', 'b', 'black', 'yellow'])

plt.scatter(X[:, 0], X[:, 1], c=np.ravel(idx), cmap=cm\_dark, s=20)

plt.scatter(centroids[:, 0], centroids[:, 1], c=np.arange(len(centroids)), cmap=cm\_dark, marker='\*', s=500)

plt.show()

