机器学习导论

编程作业：神经网络

姓名：陶威

班级：电信1604班

学号：U201613331

指导老师：王邦

# 任务一：使用Logistic回归估计马疝病的死亡率

## 实验原理：

根据现有数据对分类边界线(Decision Boundary)建立回归公式，以此进行分类。具体采用梯度上升法求最大化似然函数来求目标函数的最大值。

## 实验步骤：

1. 载入训练集和测试集数据，存储为相应向量

*with* open('data/horseColicTraining.txt', 'r') *as* f:  
 train = np.array([[float(word) *for* word *in* line.strip().split('\t')] *for* line *in* f.readlines()])  
X\_train = train[:, :-1]  
y\_train = train[:, -1]  
  
*with* open('data/horseColicTest.txt', 'r') *as* f:  
 test = np.array([[float(word) *for* word *in* line.strip().split('\t')] *for* line *in* f.readlines()])  
X\_test = test[:, :-1]  
y\_test = test[:, -1]

1. 使用随机梯度上升法建立回归模型

*def* stochastic\_gradient\_ascent(matrix, class\_labels, n\_iters=150):  
 *"""  
 随机梯度上升算法  
 """* m, n = np.shape(matrix)  
 weights = np.ones(n)  
 *for* i *in* range(n\_iters):  
 index = range(m)  
 *for* j *in* range(m):  
 alpha = 4 / (1.0 + i + j) + 0.0001 *# alpha 会随着迭代不断减小，但永远不会减小到0，因为后边还有一个常数项0.0001* rand\_index = int(np.random.uniform(0, len(index)))  
 h = sigmoid(sum(matrix[index[rand\_index]] \* weights))  
 error = class\_labels[index[rand\_index]] - h  
 weights = weights + alpha \* error \* matrix[index[rand\_index]]  
 *# del (index[rand\_index])  
 return* weights

1. 对测试集数据进行预测

*def* classify(X, weights):  
 prob = sigmoid(sum(X \* weights))  
 *return* 1 *if* prob > 0.5 *else* 0

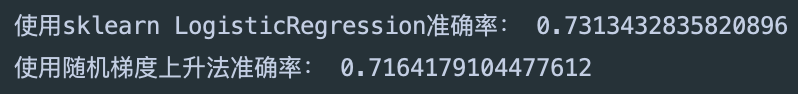
1. 计算分类准确度

train\_weights = stochastic\_gradient\_ascent(X\_train, y\_train)  
y\_predict = np.array([classify(vect, train\_weights) *for* vect *in* X\_test])  
error\_rate = sum(y\_test == y\_predict) / len(y\_test)  
print('使用随机梯度上升法准确率：', error\_rate)

1. 同时调用sklearn的LogisticRegression库函数进行结果对比

*from* sklearn.linear\_model *import* LogisticRegression  
log\_reg = LogisticRegression(solver='liblinear')  
log\_reg.fit(X\_train, y\_train)  
print('使用sklearn LogisticRegression准确率：', log\_reg.score(X\_test, y\_test))

## 实验结果：



# 任务二：使用神经网络完成新闻分类

## 实验原理：

使用多层感知机构建神经网络模型，实现对新闻文档的标签分类。

## 实验流程：

1. 读取数据集

*with* open('data/train/train\_texts.dat', 'rb') *as* f:  
 train\_text = pickle.load(f)  
  
*with* open('data/train/train\_labels.txt', 'r') *as* f:  
 y\_train = [int(line.strip()) *for* line *in* f.readlines()]

1. 提取文档特征，采用TF-IDF特征

*from* sklearn.feature\_extraction.text *import* TfidfVectorizer

vectorizer = TfidfVectorizer(max\_features=10000)  
vectors\_train = vectorizer.fit\_transform(train\_text)

1. 训练神经网络

*from* sklearn.neural\_network *import* MLPClassifier  
clf = MLPClassifier()  
clf.fit(vectors\_train, y\_train)

1. 载入测试集并预测结果，存入文件

*with* open('data/test/test\_texts.dat', 'rb') *as* f:  
 test\_text = pickle.load(f)  
vectorizer = TfidfVectorizer(max\_features=10000)  
vectors\_test = vectorizer.fit\_transform(test\_text)  
  
y\_predict = clf.predict(vectors\_test)  
*with* open('data/test/predict\_data.txt', 'w') *as* f:  
 f.write('\n'.join([str(num) *for* num *in* y\_predict]))

## 模型结构：

