电子科技大学信息与软件工程学院

实 验 报 告

学　　号　　201809162008

姓　　名　　邓萌达

（实验）课程名称　　人工智能

理论教师　　周帆

实验教师　　周帆

电子科技大学教务处制表

电 子 科 技 大 学

实　验　报　告

学生姓名：邓萌达　　**学号：**2018091620008　**指导教师：**周帆

实验地点：宿舍　　**实验时间：**2021-1-16

一、实验名称：分类任务—真假钞票鉴别

二、实验学时：4

三、实验目的：

- 掌握各种分类的使用场景和使用方法
- 掌握各种分类的理论知识

四、实验原理：

- 决策树分类
    - 决策树是一种树形结构，其中每个内部节点表示一个属性上的判断，每个分支代表一个判断结果的输出，最后每个叶节点代表一种分类结果。决策树是一种十分常用的分类方法，需要监管学习。监管学习就是给出一堆样本，每个样本都有一组属性和一个分类结果,也就是分类结果已知,那么通过学习这些样本得到一个决策树，这个决策树能够对新的数据给出正确的分类。
- KNN 分类
    - KNN 是通过测量不同特征值之间的距离进行分类。它的的思路是：如果一个样本在特征空间中的 k 个最相似(即特征空间中最邻近)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。K 通常是不大于 20 的整数。KNN 算法中，所选择的邻居都是已经正确分类的对象。该方法在定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。
- 逻辑斯蒂分类
    - Logistic 分布是一种连续性的概率分布,其分布函数和密度函数分

别为：

$$F(x) = P(X \leq x) = \frac{1}{1 + e^{-(x-\mu)/\gamma}}$$

$$f(x) = F^{'}(X \leq x) = \frac{e^{-(x-\mu)/\gamma}}{\gamma(1 + e^{-(x-\mu)/\gamma})^2}$$

  - 逻辑斯蒂分布函数是一个 s 形曲线，取值在[0, 1]之间，在远离 0 的地方函数的值会会快接近 0/1。这个性质使我们能以概率的方式来解释。

- 支持向量机 SCV 分类
  - 支持向量分类是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器。间隔最大使它有别于感知机；SVM 还包括核技巧，这使它成为实质上的非线性分类器。
  - SVM 的学习策略在于间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的核页损失函数的最小化问题。

五、实验内容：

- 分类任务—真假钞票鉴别
- 使用查准率（Precision）和查全率（Recall）指标进行评估

六、实验器材（设备、元器件）：

- MacBook Pro (macOS 16.0)
- PyCharm (2020.03)
- Python 3.7
- Anaconda3

七、实验步骤：

- 加载数据集 data_banknote_authentication.txt
- 划分数据集 train&test
- 使用不同的算法进行分类

- 决策树分类
- 逻辑斯蒂
- SVC 分类
- KNN 分类

- 使用查准率（Precision）和查全率（Recall）指标进行评估
- 分析各个算法的优点和缺点，分析得到不同的评估结果的原因

八、实验结果与分析（含重要数据结果分析或核心代码流程分析）

- 加载数据集

```
# read data
def load_data():
    data_set = pd.read_table("data_banknote_authentication.txt", header=None, sep=',')
    return data_set[0], data_set[1], data_set[2], data_set[3], data_set[4]

```

- 划分数据集

```
# using four features and split the data set
def use_four_features(x_0_set, x_1_set, x_2_set, x_3_set, y_set):
    x_set = list(zip(x_0_set, x_1_set, x_2_set, x_3_set))
    x_set = np.array(x_set)
    return train_test_split(x_set, y_set, train_size=1200)
```

- 决策树分类

```
def decision_tree(x_train, x_test, y_train, y_test):
    clf = tree.DecisionTreeClassifier()
    clf.fit(x_train, y_train)
    y_predict = clf.predict(x_test)
    print(metrics.precision_score(y_test, y_predict))
    print(metrics.recall_score(y_test, y_predict))
```

- 逻辑斯蒂

```
def logistics_classifier(x_train, x_test, y_train, y_test):
    logistics = linear_model.LogisticRegression()
    logistics.fit(x_train, y_train)
    y_predict = logistics.predict(x_test)
    print(metrics.precision_score(y_test, y_predict))
    print(metrics.recall_score(y_test, y_predict))
```

- SCV 分类

```
def svc_classifier(x_train, x_test, y_train, y_test):
    clf = SVC(C=0.8, kernel='rbf', gamma=2, decision_function_shape='ovr')
    clf.fit(x_train, y_train)
    y_predict = clf.predict(x_test)
    print(metrics.precision_score(y_test, y_predict))
    print(metrics.recall_score(y_test, y_predict))
```

- KNN 分类

```
50 def knn_classifier(x_train, x_test, y_train, y_test):
51     knn = neighbors.KNeighborsClassifier(n_neighbors=2, weights='uniform', algorithm='auto')
52     knn.fit(x_train, y_train)
53     y_predict = knn.predict(x_test)
54     print(metrics.precision_score(y_test, y_predict))
55     print(metrics.recall_score(y_test, y_predict))
```

- 结果：
    - 决策树分类的查准率（Precision）和查全率（Recall）

```
decision_tree:
0.9863013698630136
0.972972972972973
```

    - 逻辑斯蒂分类的查准率（Precision）和查全率（Recall）

```
logistics_classifier:
0.9866666666666667
1.0
```

    - SCV 分类的查准率（Precision）和查全率（Recall）

```
svc_classifier:
1.0
1.0
```

    - KNN 分类的查准率（Precision）和查全率（Recall）

```
knn_classifier:
1.0
1.0
```

- 各个算法的分析见第九（总结）部分

九、总结及心得体会：

- 这次实验完成了对真假钞票的鉴别，采用了决策树分类，逻辑斯蒂分类，SCV 分类，KNN 分类，并计算查准率（Precision）和查全率（Recall），完成了实验要求，达到了实验目的。
- 在本次实验中使用了四种分类方法，其中基于向量机的查准率和查全率都为 1，逻辑斯蒂分类的查准率没有到达 1，决策树分类的查准率和查全率都未到达 1。但是，再多次重复运行之后，决策树分类有到达过查准率和查全率都为 1 的情况，逻辑斯蒂则一直有较小的错误率。KNN 和 SCV 都表现稳定在全部查准和查全。
- 由结果可知，SCV、KNN 算法对于此数据集的训练结果都很理想。事实上，KNN

算法简单，理论成熟，并且可用于非线性分类，由于 KNN 方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属的类别，因此对于类域的交叉或重叠较多的待分类样本集来说，KNN 方法较其他方法更为适合。此外，KNN 算法比较适用于样本容量比较大的类域的自动分类，而那些样本容量比较小的类域采用这种算法比较容易产生误分类情况。但是，当数据量大的时候，由于该算法的计算复杂度特别高，所以分类速度会非常满，尤其是特征很多时，由于该算法需要大量的内存，空间也复杂度高，此外，该算法是 lazy learning 方法，基本上不学习，导致预测时速度比起逻辑回归之类的算法慢。这也就是为什么 KNN 在我们这个数据集上表现优秀，如果数据集过大，KNN 将不再适合。

- SVM 相比于 KNN 最大的优点就是速度快，它是一种有坚实理论基础的新颖的小样本学习方法。它基本上不涉及概率测度及大数定律等,因此不同于现有的统计方法。从本质上看,它避开了从归纳到演绎的传统过程,实现了高效的从训练样本到预报样本的“转导推理”,大大简化了通常的分类和回归等问题。然而，他的缺点和 KNN 一样，SVM 算法对大规模训练样本难以实施，由于 SVM 是借助二次规划来求解支持向量，而求解二次规划将涉及 m 阶矩阵的计算（m 为样本的个数），当 m 数目很大时该矩阵的存储和计算将耗费大量的机器内存和运算时间。而且，它在解决多分类的问题时也不如 KNN。
- 决策树是在已知各种情况发生概率的基础上，通过构成决策树来求取净现值的期望值大于等于零的概率，评价项目风险，判断其可行性的决策分析方法，是直观运用概率分析的一种图解法。它是一种十分常用的分类方法，是一种监督学习。所谓监督学习就是给定一堆样本，每个样本都有一组属性和一个类别，这些类别是事先确定的，那么通过学习得到一个分类器，这个分类器能够对新出现的对象给出正确的分类。这样的机器学习就被称之为监督学习。决策树的缺点在于对连续性的字段比较难预测，)当类别太多时，错误可能就会增加的比较快。
- 逻辑回归是一个非常经典的算法，使用 sigmod 函数加上线性回归，虽然他是回归算法，但是由于 sigmod 函数的特性，经常被用于分类之中，尤其是二分类。此外，该算法不仅可预测出类别，还能得到该预测的概率，这对一些利

用概率辅助决策的任务很有用。然而，因为它本质上是一个线性的分类器，所以处理不好特征之间相关的情况，特征空间很大时，性能不好。最重要的一点是，由于 sigmod 函数的特性， 他十分容易欠拟合，导致精度不高。

十、对本实验过程及方法、手段的改进建议：

- 这次实验选择的数据选择较为清晰，可以容易的使用各个基本的分类算法来进行分类，并且得到较好的结果。
- 但是这次实验中，仅给出了一组数据，并且让学生们自行选择算法，这样的话会造成不同算法对不同数据集的区别不明显，学生不能很好的体会到算法的针对性，建议在今后的实验中，可以给出多组数据，并且让学生使用不同的算法给出截然不同的分类效果，让学生的印象更加深刻。

报告评分：

指导教师签字：