

**《机器学习》**

**课程设计期末报告**

**报告题目：经典数据集上的KNN、决策树与朴素贝叶斯权重**

**学院名称： 计算机与人工智能学院**

**年 级： 2022级**

**授课教师：**  **胡 节**

**学 生： 陈垠作**

**二〇二四年十一月**

# 引言

本篇报告主要研究三种经典的分类算法——K近邻（KNN）、决策树（Decision Tree）和朴素贝叶斯（Naive Bayes），并在多个公开数据集上进行对比实验，旨在深入理解算法原理、性能差异及适用场景，为实际应用提供理论和实践参考。其中公开数据集包含Iris，Wine，Breast Cancer，Digits数据集。

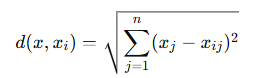
# 相关工作

近年来，分类算法得到了深入研究和广泛应用。KNN算法因其简单直观，在推荐系统和图像识别中表现出色 [1]。决策树因其强大的解释性和良好的分类效果，广泛应用于金融风控和医学诊断领域 [2]。朴素贝叶斯基于贝叶斯定理，适合文本分类和大规模数据处理 [3]。本文综合比较三者性能，补充模型权重可视化及特征重要性分析。

# 算法原理

## K近邻（KNN）

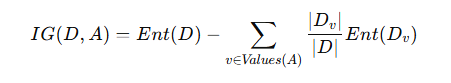
KNN是一种基于实例的学习方法。分类时，计算待分类样本与训练集中所有样本的距离，选择距离最近的K个邻居，通过多数表决确定类别。其核心公式为欧氏距离：



优点是无训练过程，简单易用；缺点是计算量大，易受噪声影响。

## 决策树（Decision Tree）

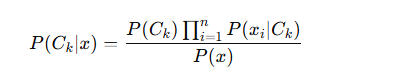
决策树使用树形结构将样本划分为不同类别。通过信息增益或基尼指数选择最优特征进行节点分裂。信息增益定义为：



其中，Ent(D)Ent(D)Ent(D) 是数据集D的信息熵。决策树可解释性强，但易过拟合。

## 朴素贝叶斯（Naive Bayes）

朴素贝叶斯基于贝叶斯定理，假设特征条件独立，计算后验概率：



选取后验概率最大的类别。算法高效，适合高维数据，但独立假设限制其性能。

# 数据集与预处理

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集 | 特征数 | 类别数 | 样本数 |
| Iris | 4 | 3 | 150 |
| Wine | 13 | 3 | 178 |
| Breast Cancer | 30 | 2 | 569 |
| Digits | 64 | 10 | 1797 |

数据集来源于 `scikit-learn`，划分比例为训练集70%、测试集30%。

# 实验设计

## 实验环境

系统：Windows 10 64位

编程语言：Python 3.12

开发工具：PyCharm

依赖库：numpy、pandas、scikit-learn、matplotlib、pickle

## 数据预处理

缺失值填充（Titanic使用众数填充）

类别变量编码（独热编码）

特征归一化（Min-Max归一化，确保KNN距离计算有效）

## 实现流程

### 载入训练与测试数据

### 数据预处理

### 训练KNN、决策树、朴素贝叶斯模型

### 保存模型权重（pickle格式）

### 载入模型权重进行预测

### 计算准确率与性能指标

### 结果汇总与分析

# 实验结果与分析

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集 | KNN准确率 | 决策树准确率 | 朴素贝叶斯准确率 |
| Iris | 0.956 | 0.911 | 0.911 |
| Wine | 0.648 | 0.907 | 1.000 |
| Breast Cancer | 0.918 | 0.953 | 0.936 |
| Digits | 0.987 | 0.822 | 0.787 |

## 结果分析

### Iris数据集

KNN算法表现最佳，准确率达95.6%，决策树和朴素贝叶斯相同，均为91.1%，说明简单距离度量方法对该数据集效果显著。

### Wine数据集

朴素贝叶斯达到100%准确率，表现最好，说明其条件独立假设在此数据上符合较好；KNN表现较差（64.8%），可能受数据高维影响。

### Breast Cancer数据集

决策树最高（95.3%），KNN和朴素贝叶斯表现接近，表明决策树在医疗诊断类数据中具有较强分类能力。

### Digits数据集

KNN准确率最高（98.7%），显示出KNN在图像数字识别中优秀的性能；决策树和朴素贝叶斯准确率相对较低，可能因样本特征复杂和分布差异大。

## 权重获取

对于权重的获取，本代码保存在pkl中，但是介于pkl格式观察麻烦，为了方便查看写了Read\_pkl.py来方便查看，且直接贴入报告

====== breast\_cancer 数据集模型权重信息 ======

KNN 模型对象: <models.knn.KNNClassifier object at 0x0000029E0B9D9A00>

X\_train 形状: (398, 30)

y\_train 形状: (398,)

决策树模型对象: <models.decision\_tree.DecisionTreeClassifier object at 0x0000029E0B9D9FA0>

特征重要性: {'worst radius<=16.7950': 0.06666666666666667, 'worst concave points<=0.1366': 0.06666666666666667, 'symmetry error<=0.0166': 0.06666666666666667, 'radius error<=0.5470': 0.06666666666666667, 'worst texture<=29.7550': 0.06666666666666667, 'mean texture<=23.2000': 0.06666666666666667, 'mean radius<=12.5100': 0.06666666666666667, 'mean perimeter<=86.5450': 0.06666666666666667, 'worst texture<=25.6200': 0.06666666666666667, 'mean smoothness<=0.1226': 0.06666666666666667, 'worst area<=817.1000': 0.06666666666666667, 'mean fractal dimension<=0.0609': 0.06666666666666667, 'mean concavity<=0.0721': 0.06666666666666667, 'mean texture<=19.8300': 0.06666666666666667, 'mean compactness<=0.0592': 0.06666666666666667}

朴素贝叶斯模型对象: <models.naive\_bayes.NaiveBayesClassifier object at 0x0000029E0DDE77A0>

类别: [0 1]

均值 (mean):

类别 0: [1.74477703e+01 2.16473649e+01 1.15229459e+02 9.73026351e+02

1.02433041e-01 1.45420676e-01 1.61745203e-01 8.74699324e-02

1.92947297e-01 6.26747973e-02 5.71679054e-01 1.15816149e+00

4.07153378e+00 6.68486486e+01 6.52691216e-03 3.22892365e-02

4.24848649e-02 1.48563311e-02 2.05538986e-02 3.97426351e-03

2.11802703e+01 2.94818919e+01 1.41713851e+02 1.42459865e+03

1.45122432e-01 3.88118919e-01 4.73537162e-01 1.85297770e-01

3.30266892e-01 9.26471622e-02]

类别 1: [1.21083240e+01 1.79764800e+01 7.77688800e+01 4.60016800e+02

9.18561600e-02 7.81153200e-02 4.50525028e-02 2.47113760e-02

1.73120000e-01 6.27301600e-02 2.85498000e-01 1.22155520e+00

2.00520440e+00 2.12644120e+01 7.25062000e-03 2.12606760e-02

2.60873064e-02 9.56348400e-03 2.07572360e-02 3.70663240e-03

1.33576440e+01 2.36462800e+01 8.68007200e+01 5.57114800e+02

1.24644000e-01 1.79648200e-01 1.63371588e-01 7.25632040e-02

2.70838800e-01 7.92458000e-02]

方差 (var):

类别 0: [9.56097408e+00 1.64480302e+01 4.44059192e+02 1.16026203e+05

1.46388686e-04 2.93242068e-03 5.47027202e-03 1.13322764e-03

8.25088222e-04 5.97221290e-05 6.86552178e-02 2.07596687e-01

4.10939656e+00 1.70854057e+03 8.19130465e-06 3.64450951e-04

5.01027711e-04 3.38382273e-05 1.14453569e-04 4.12807379e-06

1.81534878e+01 3.14568397e+01 8.60041926e+02 3.40704084e+05

4.46504108e-04 3.15206611e-02 3.61106667e-02 2.23490015e-03

6.00449619e-03 5.16988193e-04]

类别 1: [3.07540636e+00 1.74153492e+01 1.34852384e+02 1.73386931e+04

1.54717940e-04 1.10139203e-03 2.24812077e-03 2.42462223e-04

6.15295000e-04 4.95901176e-05 1.34847173e-02 3.42400753e-01

6.07603217e-01 8.54283350e+01 9.57904143e-06 2.94429738e-04

1.38040630e-03 3.36631920e-05 5.38043641e-05 1.09831411e-05

3.84861796e+00 3.23097930e+01 1.80943628e+02 2.61551303e+04

3.96654797e-04 8.25793315e-03 2.08733070e-02 1.23263511e-03

1.86967217e-03 2.03440453e-04]

先验概率 (priors):

类别 0: 0.37185929648241206

类别 1: 0.628140703517588

====== iris 数据集模型权重信息 ======

KNN 模型对象: <models.knn.KNNClassifier object at 0x0000029E0DDE77A0>

X\_train 形状: (105, 4)

y\_train 形状: (105,)

决策树模型对象: <models.decision\_tree.DecisionTreeClassifier object at 0x0000029E0DE338C0>

特征重要性: {'petal length (cm)<=2.4500': 0.16666666666666666, 'petal width (cm)<=1.5500': 0.16666666666666666, 'sepal width (cm)<=2.2500': 0.16666666666666666, 'sepal length (cm)<=5.5000': 0.16666666666666666, 'sepal length (cm)<=6.1000': 0.16666666666666666, 'sepal width (cm)<=3.1000': 0.16666666666666666}

朴素贝叶斯模型对象: <models.naive\_bayes.NaiveBayesClassifier object at 0x0000029E104E3290>

类别: [0 1 2]

均值 (mean):

类别 0: [4.98857143 3.42571429 1.48571429 0.24 ]

类别 1: [5.94857143 2.73142857 4.23714286 1.30857143]

类别 2: [6.68285714 3.00857143 5.63142857 2.06857143]

方差 (var):

类别 0: [0.10329796 0.17391021 0.02293878 0.00925714]

类别 1: [0.24078367 0.08558367 0.21147755 0.03564082]

类别 2: [0.42484898 0.1173551 0.32272653 0.06386939]

先验概率 (priors):

类别 0: 0.3333333333333333

类别 1: 0.3333333333333333

类别 2: 0.3333333333333333

====== wine 数据集模型权重信息 ======

KNN 模型对象: <models.knn.KNNClassifier object at 0x0000029E104E3290>

X\_train 形状: (124, 13)

y\_train 形状: (124,)

决策树模型对象: <models.decision\_tree.DecisionTreeClassifier object at 0x0000029E0DE338C0>

特征重要性: {'flavanoids<=1.5750': 0.2, 'hue<=0.8980': 0.2, 'alcohol<=13.1650': 0.2, 'alcohol<=13.0200': 0.2, 'magnesium<=88.0000': 0.2}

朴素贝叶斯模型对象: <models.naive\_bayes.NaiveBayesClassifier object at 0x0000029E1058BD40>

类别: [0 1 2]

均值 (mean):

类别 0: [1.37304878e+01 1.94707317e+00 2.44975610e+00 1.71024390e+01

1.06634146e+02 2.82853659e+00 2.94024390e+00 3.01707317e-01

1.85121951e+00 5.56780488e+00 1.05097561e+00 3.08853659e+00

1.11280488e+03]

类别 1: [1.22424e+01 1.96260e+00 2.23280e+00 2.05240e+01 9.51400e+01 2.25360e+00

2.04680e+00 3.50800e-01 1.71220e+00 2.96080e+00 1.05892e+00 2.80220e+00

5.31260e+02]

类别 2: [1.30745455e+01 3.20090909e+00 2.45424242e+00 2.15606061e+01

9.92727273e+01 1.68757576e+00 7.87575758e-01 4.46363636e-01

1.13878788e+00 7.36272724e+00 6.73030303e-01 1.69060606e+00

6.24393939e+02]

方差 (var):

类别 0: [2.02950983e-01 3.95762166e-01 6.05975025e-02 7.29535991e+00

1.16280785e+02 1.14880786e-01 1.37626771e-01 5.39464704e-03

1.46386319e-01 1.48304152e+00 1.26624638e-02 1.08866152e-01

3.97256692e+04]

类别 1: [2.70046241e-01 1.10783524e+00 7.88841610e-02 1.05854240e+01

3.47600400e+02 2.97211041e-01 3.66681761e-01 1.23193610e-02

3.59241161e-01 7.50035361e-01 3.96851546e-02 2.28521161e-01

2.57562324e+04]

类别 2: [2.61655097e-01 8.79135538e-01 3.18062453e-02 4.78420569e+00

1.02865014e+02 1.47394124e-01 1.02339579e-01 1.45322324e-02

2.12774289e-01 5.87018938e+00 1.39726364e-02 8.49269064e-02

1.35632691e+04]

先验概率 (priors):

类别 0: 0.33064516129032256

类别 1: 0.4032258064516129

类别 2: 0.2661290322580645

# 代码说明与可执行性

代码结构清晰，模块化设计，易于维护

所有依赖库均为主流公开库，安装简单

使用pickle保存与加载模型，便于模型复用

详细注释说明每一步，方便理解与二次开发

# 参考文献

[1] Tom Mitchell, Machine Learning, McGraw Hill, 1997.  
[2] L. Breiman et al., Classification and Regression Trees, Wadsworth, 1984.  
[3] T. Hastie et al., The Elements of Statistical Learning, Springer, 2009.  
[4] Pedregosa et al., Scikit-learn: Machine Learning in Python, JMLR, 2011.