

DataMining_homework3 实验报告

姓名：张映雪

学号：2120171101

小组：第 6 组

目录

1. 实验概述	3
2. 实验环境	3
3. 实验过程与分析	4
3.1 数据集介绍	4
3.2 分类	4
3.2.1 SVM 分类器	4
3.2.2 决策树分类器	6
3.3 聚类	7
3.3.1 K-means 聚类	7
3.3.2 DBSCAN 密度聚类	8
4 实验总结	9

1. 实验概述

1.1 实验所用数据集

- Kaggle/Titanic

1.2 实验要求

- 使用分类模型（至少 2 个）对数据集进行挖掘；
- 对挖掘结果进行可视化，并解释其意义；
- 使用聚类方法（至少 2 种）对数据集进行分析；
- 对挖掘结果进行可视化，并解释其意义。

1.3 实验涉及算法

- 分类：SVM、决策树
- 聚类：K-means、DBSCAN 密度聚类

2. 实验环境

- 使用 python3.6
- 所用的包： numpy、pandas、matplotlib、sklearn 等

3. 实验过程与分析

3.1 数据集介绍

本次实验选用数据为 kaggle 竞赛中 Titanic 题目中的数据集 train.csv

	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerID	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, Mrs	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen, Mrs	female	26	0	0	STON/O2	7.925		S
5	4	1	1	Futrelle, Mrs	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy, Mr	male	54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, Mr	male	2	3	1	349909	21.075		S
10	9	1	3	Johnson, Mrs	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, Mrs	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstrom, Mrs	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, Mrs	female	58	0	0	113783	26.55	C103	S
14	13	0	3	Saunders, Mr	male	20	0	0	A/5. 2151	8.05		S
15	14	0	3	Andersson, Mr	male	39	1	5	347082	31.275		S
16	15	0	3	Vestrom, Mrs	female	14	0	0	350406	7.8542		S
17	16	1	2	Hewlett, Mrs	female	55	0	0	248706	16		S
18	17	0	3	Rice, Master	male	2	4	1	382652	29.125		Q
19	18	1	2	Williams, Mr	male		0	0	244373	13		S
20	19	0	3	Vander Planck, Mrs	female	31	1	0	345763	18		S
21	20	1	3	Masselmani, Mrs	female		0	0	2649	7.225		C
22	21	0	2	Fynney, Mr	male	35	0	0	239865	26		S

由上图可见，数据集共有 12 列，“survived”是最终的分类目标，把该列视为标注。其余各列中 PassengerID, Name 对判断“是否存活”没有什么意义，所以分类是不采用这两列数据。经实验发现“Ticket”、“Embarked”对在 SVM，决策树分类中反而会起到降低分类效果的作用，因此也不选用此两列。因此最终用于分类聚类的列数为“Pclass, Sex, Age, SibSp, Fare, Cabin”。

3.2 分类

3.2.1 SVM 分类器

经过尝试，构建 SVM 分类器时，kernel 选用‘rbf’的效果最高，分类器的构建

```
def svm_classify(x_train, x_test, y_train, y_test):  
    svm_classifier = svm.SVC(C=0.5, kernel='rbf', gamma=0.5)  
    svm_classifier.fit(x_train, y_train)  
    print("svm分类准确率：" + str(svm_classifier.score(x_test, y_test)))  
    return svm_classifier
```

部分源码如下：

分类结果为

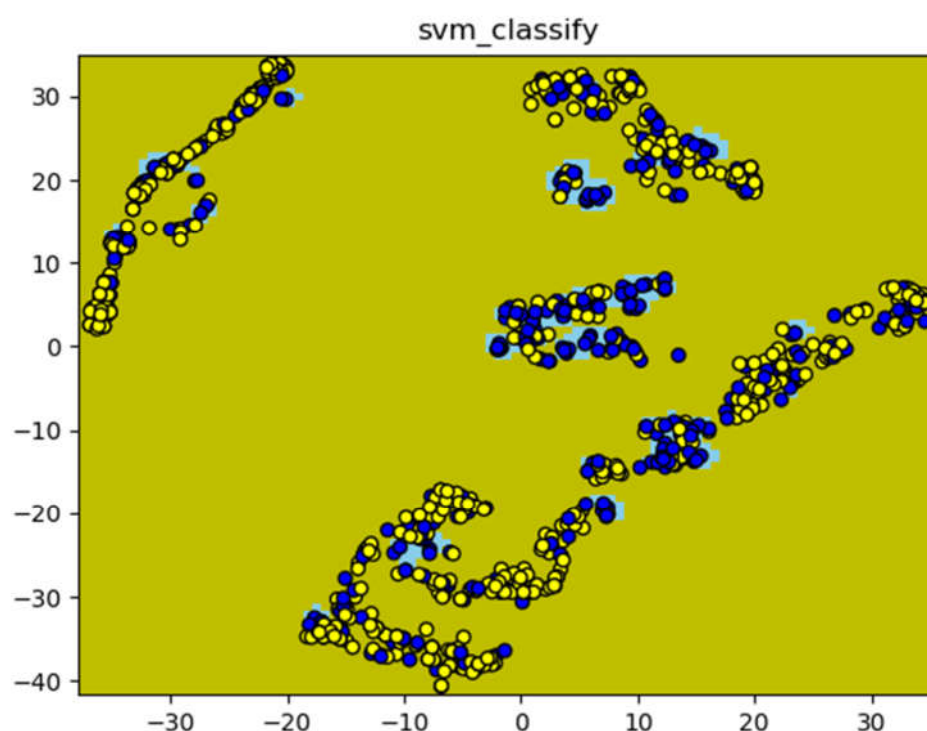
svm分类准确率:0.6

可视化：

在可视化前，首先将数据降为二维，将数据降维后的分类准确率反而有所提升，为：

svm分类准确率:0.7

将其可视化



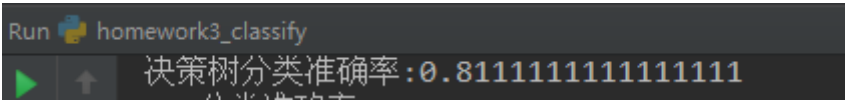
如上图，以“边”绘出分界面，点代表样本点。两种不同颜色（浅蓝色、土黄色）的被“边”围起来的区域代表的是该 SVM 分类器分出的两种类别。两种不同颜色的点代表两类样本点的真实分布，深蓝色对应浅蓝色，亮黄色对应土黄色，与相应颜色重合的样本点即该分类器可分类正确的样本点。

3.2.2 决策树分类器

构建决策树分类器时，分别尝试了最大深度 4、5、6、7 时的效果，最大深度为 7 时分类效果最好。分类器的构建部分源码如下：


```
def decision_tree(x_train, x_test, y_train, y_test, create_graph):  
    |  
    dec_classifier = tree.DecisionTreeClassifier(max_depth=7)  
    dec_classifier.fit(x_train, y_train)  
  
    print("决策树分类准确率:" + str(dec_classifier.score(x_test, y_test)))  
  
    if create_graph == True:  
        dot_data = tree.export_graphviz(dec_classifier, out_file=None)  
        graph = pydotplus.graph_from_dot_data(dot_data)  
        graph.write_png("decision_tree.png")  
    return dec_classifier
```

分类效果如下，可达 0.81



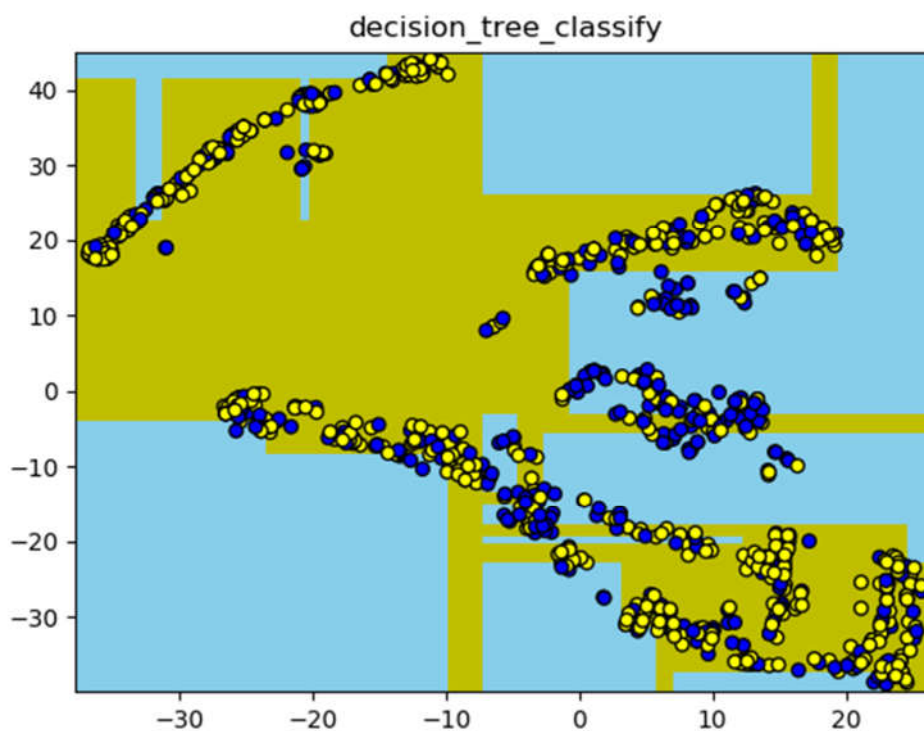
```
Run homework3_classify  
决策树分类准确率:0.8111111111111111
```

数据可视化处理与 SVM 分类器的方式一样，同样将数据降维为二维，再行处理，但由于数据降维后损失了很多信息，决策树分类器的分类效果下降很多，降到了 0.6,如下所示：



```
决策树分类准确率:0.6
```

可视化如下，图中各部分含义与 SVM 分类器一致，此处不再赘述：



3.3 聚类

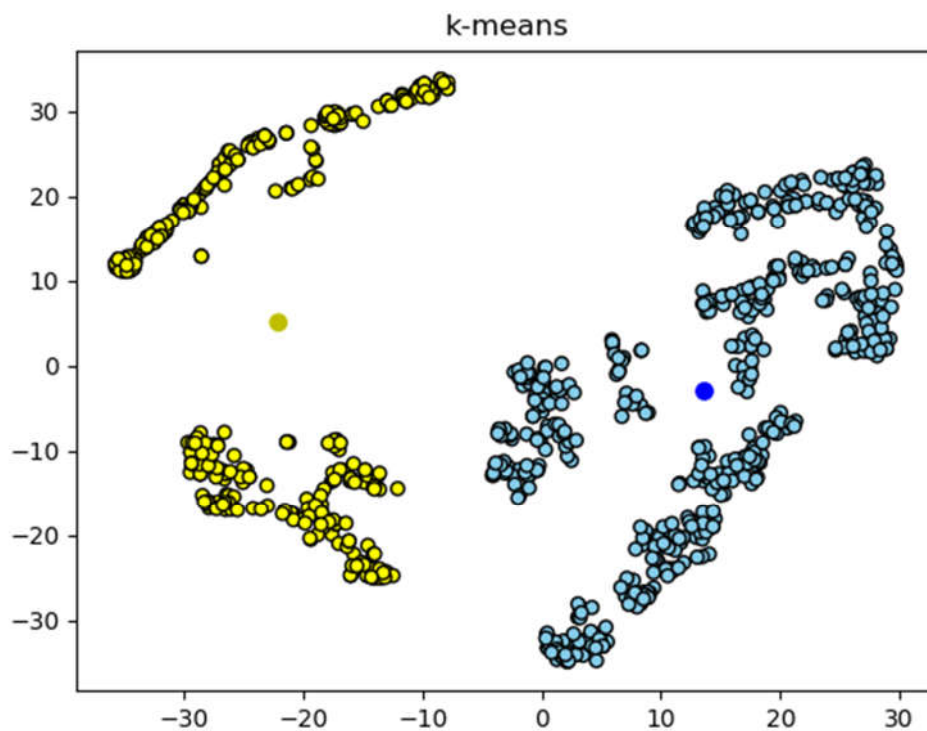
3.3.1 K-means 聚类

K-means 是机器学习中一个比较常用的算法，属于无监督学习算法，其常被用于数据的聚类，只需为它指定簇的数量即可自动将数据聚合到多类中，相同簇中的数据相似度较高，不同簇中数据相似度较低。

同样的，为了聚类的可视化，首先要把数据降维为 2 维。本次实验中，根据 label 制定簇为 2，K-means 核心源码如下：

```
k_means = KMeans(init="k-means++", n_clusters=2, random_state=28)
k_means.fit(X_tsne)
y_km = k_means.predict(X_tsne)
km_center = k_means.cluster_centers_
```

可视化如下：

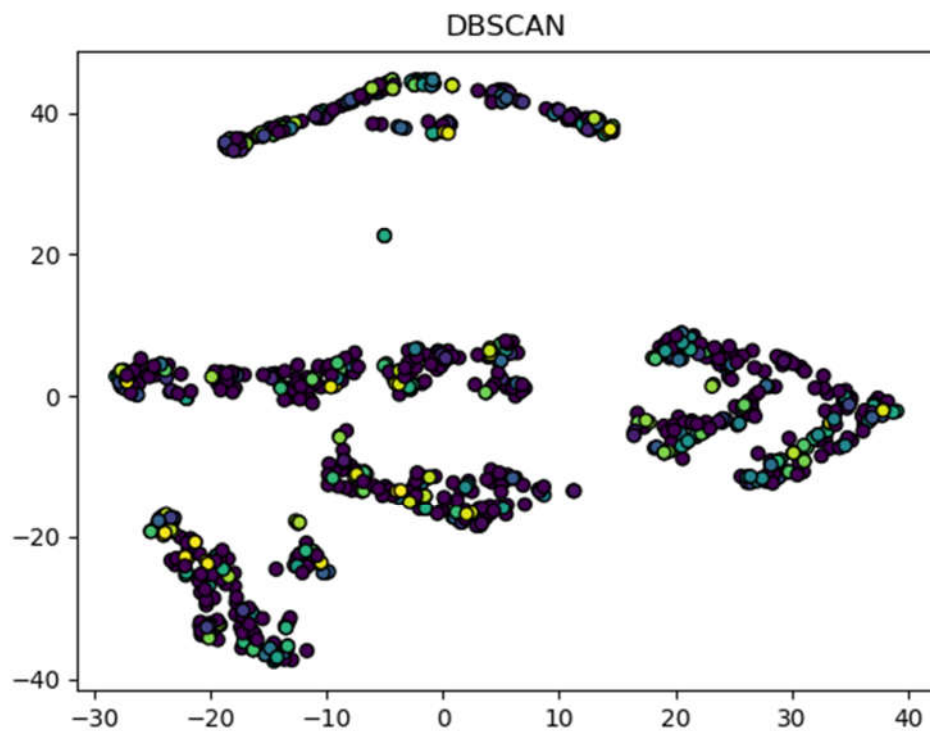


3.3.2 DBSCAN 密度聚类

DBSCAN(Density-Based Spatial Clustering of Applications with Noise, 具有噪声的基于密度的聚类方法)是一种很典型的密度聚类算法。但本次数据集数据量小且并不稠密, 实际上可能并不是很合适的聚类方法, 但出于想熟悉该算法并且探索一下的目的, 选择了该算法。该算法的调参比 k-means 复杂一些, 不同的参数组合对结果有较大影响。在本次实验中, 设置 `eps` 为 0.4, `min_samples` 为 2。源码如下:

```
# 密度聚类
y_db = DBSCAN(eps=0.4, min_samples=2).fit_predict(X_tsne)
cm_light = mpl.colors.ListedColormap(['yellow', 'skyblue'])
cm2 = mpl.colors.ListedColormap(['y', 'blue'])
```

可视化如下:



我们知道，DBSCAN 聚类算法可对任意形状的数据集进行聚类，但该数据集数据量小且数据不够稠密，并不是很适用。

4 实验总结

经过本次数据分类和聚类的实验，体会到了不同分类算法和聚类算法的优点和缺点，在课下会继续学习和探索不同数据集下分类聚类的技巧和经验。