Python与金融数据挖掘(16)

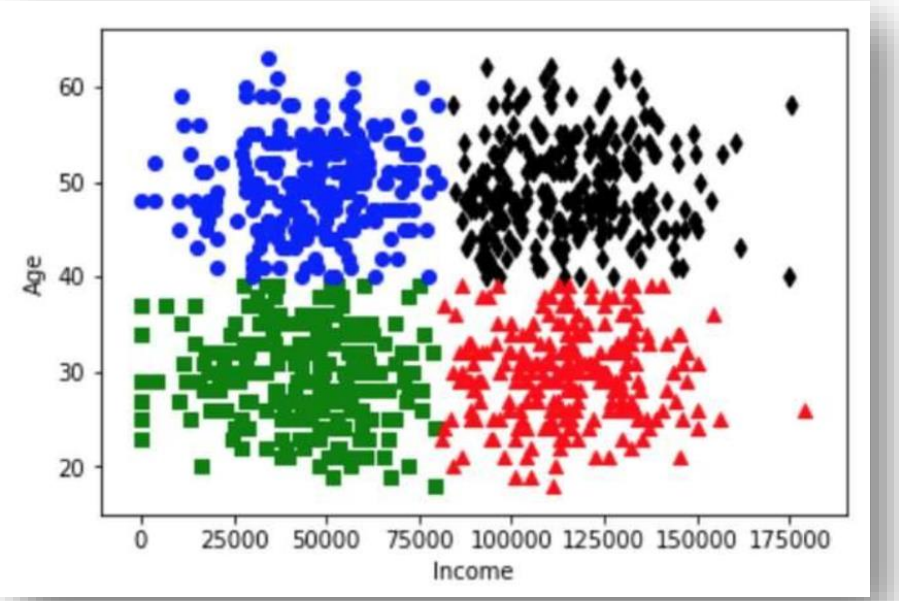
文欣秀

wenxinxiu@ecust.edu.cn

机器学习分类

- 有监督学习（分类、回归）
- 无监督学习（聚类、降维）
- 强化学习
- 半监督学习

客户类型聚类分析



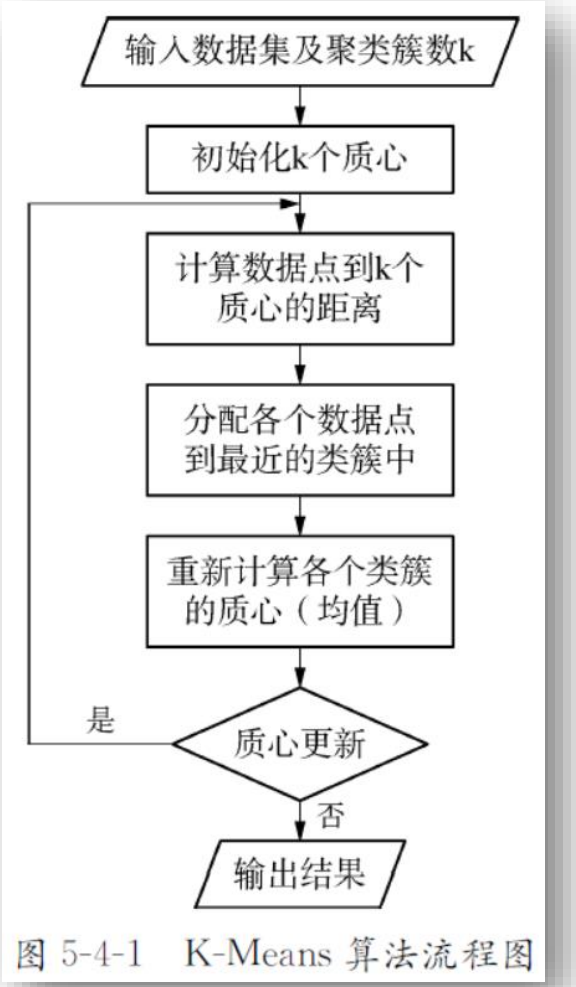
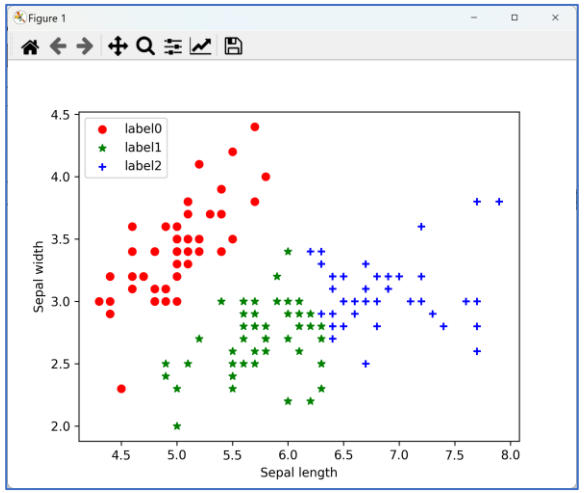
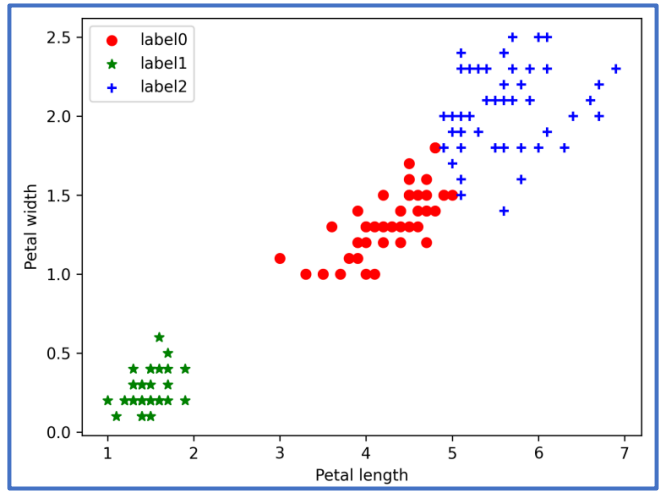


图 5-4-1　K-Means 算法流程图

鸢尾花聚类问题

	A	B	C	D
1	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width
2	5.1	3.5	1.4	0.2
3	4.9	3	1.4	0.2
4	4.7	3.2	1.3	0.2
5	4.6	3.1	1.5	0.2
6	5	3.6	1.4	0.2
7	5.4	3.9	1.7	0.4
8	4.6	3.4	1.4	0.3
9	5	3.4	1.5	0.2
10	4.4	2.9	1.4	0.2
11	4.9	3.1	1.5	0.1
12	5.4	3.7	1.5	0.2
13	4.8	3.4	1.6	0.2





K- Means聚类算法

Scikit-learn的Cluster类提供聚类分析的方法：

- 模型初始化

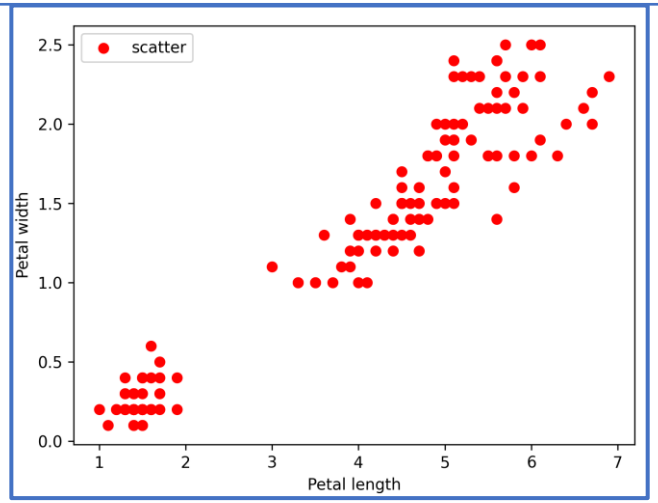
```
kmeans=Kmeans(n_clusters)    #参数为簇的个数
```

- 模型学习

```
kmeans. fit(X)                    #参数为样本二维数组
```

鸢尾花问题K- Means模型（1）

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans
import pandas as pd   #导入模块
iris = pd.read_csv("iris.csv")
X = iris.loc[:,['Petal_Length', 'Petal_Width']] #读出数据
plt.scatter(X['Petal_Length'], X['Petal_Width'], c = "red", marker='o', label='scatter')
plt.xlabel('Petal length'); plt.ylabel('Petal width')
plt.legend(loc=2); plt.show()
```

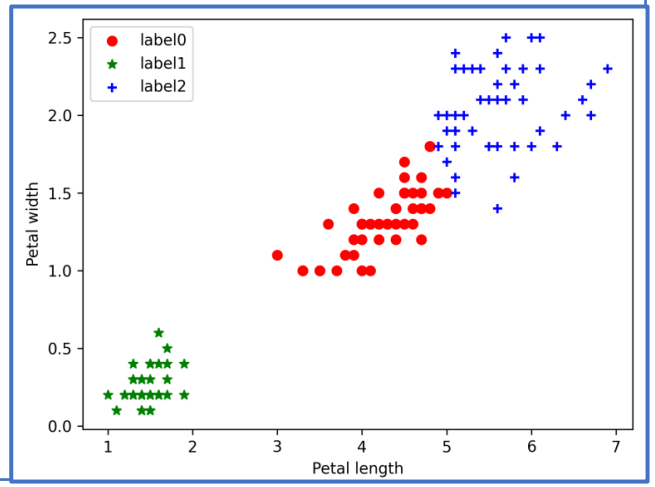


鸢尾花问题K- Means模型（2）

```
estimator = KMeans(n_clusters=3)#模型初始化
estimator.fit(X) #模型学习
label_pred = estimator.labels_ #获取聚类标签
x0 = X[label_pred == 0]
x1 = X[label_pred == 1]
x2 = X[label_pred == 2]
print(x0)
print(x1)
print(x2)
```

鸢尾花问题K- Means模型（3）

```
plt.scatter(x0['Petal_Length'], x0['Petal_Width'], c = "red", marker='o', label='label0')

plt.scatter(x1['Petal_Length'], x1['Petal_Width'], c = "green", marker='*', label='label1')

plt.scatter(x2['Petal_Length'], x2['Petal_Width'], c = "blue", marker='+', label='label2')

plt.xlabel('Petal length')

plt.ylabel('Petal width')

plt.legend(loc=2)

plt.show()
```



鸢尾花数据集获取

```
>>> from sklearn import datasets     # 导入数据集包
>>> dir (datasets)                   # 查看数据集
>>> iris = datasets.load_iris()
>>> X = iris['data']
>>> print(X)
```

导入数据的函数名称	对应的数据集
~~load_boston()~~	~~波士顿房价数据集~~
load_breast_cancer()	乳腺癌数据集
load_iris()	鸢尾花数据集
load_diabetes()	糖尿病数据集
load_digits()	手写数字数据集
load_linnerud()	体能训练数据集
load_wine()	红酒品类数据集

鸢尾花问题K- Means模型（1）

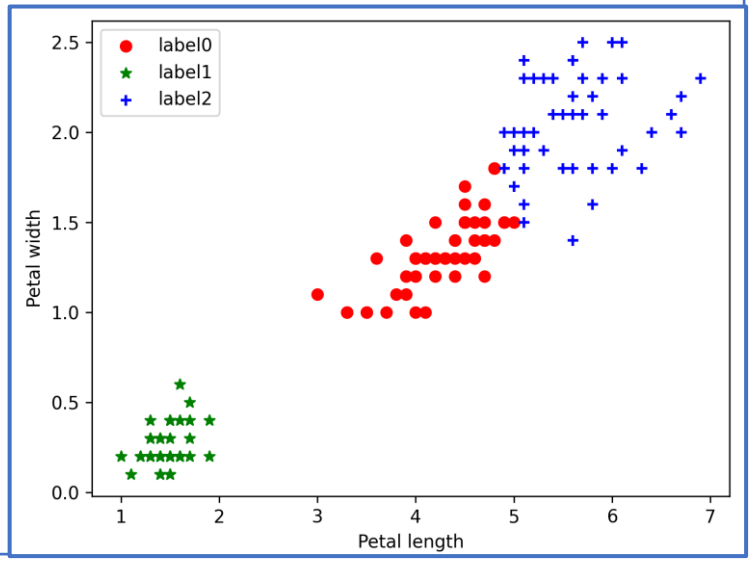
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import datasets      # 导入数据集包
iris = datasets.load_iris()       # 加载数据据集
X = iris['data']                  #读出数据
```

鸢尾花问题K- Means模型（2）

```
estimator = KMeans(n_clusters=3)#模型初始化
estimator.fit(X) #模型学习
label_pred = estimator.labels_ #获取聚类标签
x0 = X[label_pred == 0]
x1 = X[label_pred == 1]
x2 = X[label_pred == 2]
```

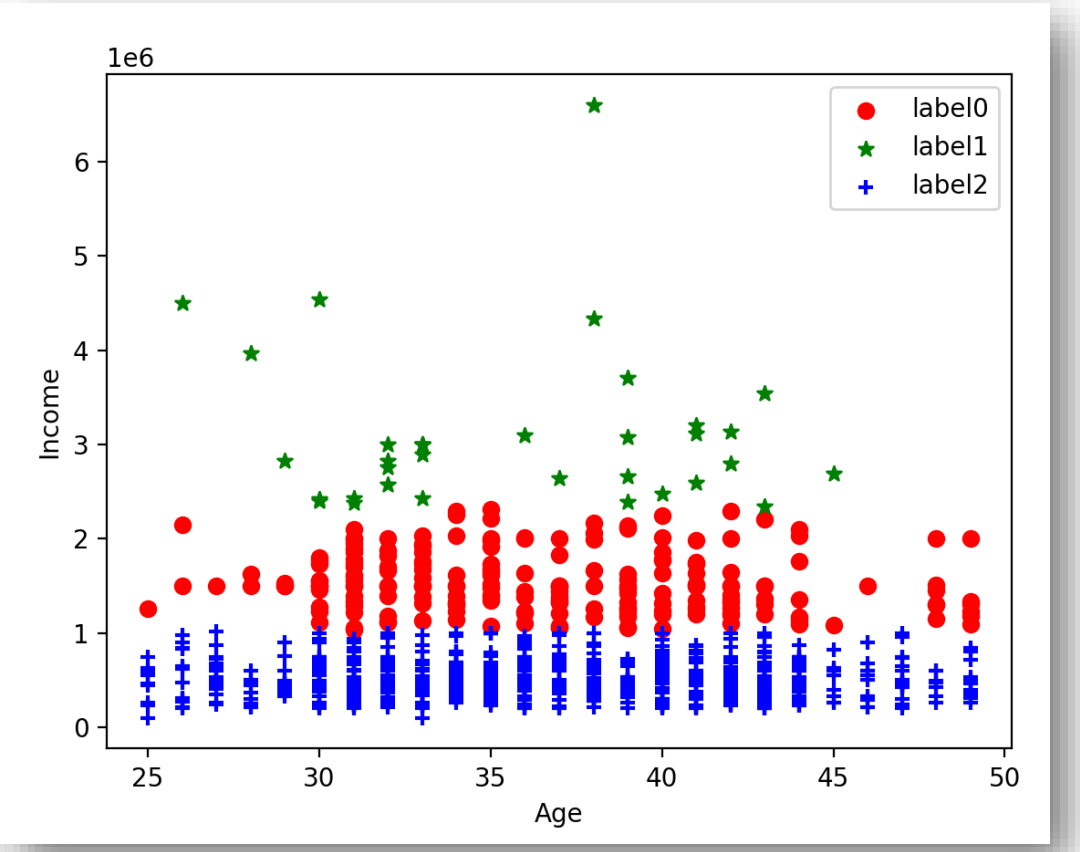
鸢尾花问题K- Means模型（3）

```
plt.scatter(x0[:,2], x0[:,3], c = "red", marker='o', label='label0')

plt.scatter(x1[:,2], x1[:,3], c = "green", marker='*', label='label1')

plt.scatter(x2[:,2], x2[:,3], c = "blue", marker='+', label='label2')

plt.xlabel('Petal length')

plt.ylabel('Petal width')

plt.legend(loc=2)

plt.show()
```



客户类型聚类分析

	A	B
1	收入	年龄
2	503999	46
3	452766	36
4	100000	33
5	100000	25
6	258000	35
7	933333	31
8	665000	40
9	291332	38
10	259000	45



客户类型聚类（1）

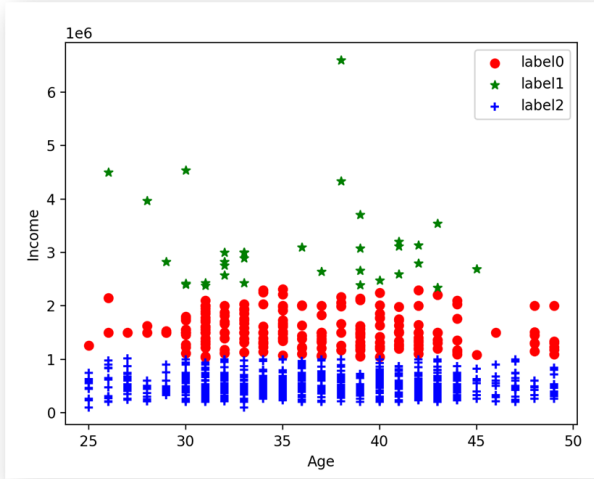
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from warnings import simplefilter
simplefilter(action='ignore', category=FutureWarning)
c = pd.read_csv("client.csv",encoding="utf-8")
```

客户类型聚类（2）

```
X=c.iloc[:,0:2].values
estimator = KMeans(n_clusters=3)#模型初始化
estimator.fit(X) #模型学习
label_pred = estimator.labels_ #获取聚类标签
x0 = X[label_pred == 0]
x1 = X[label_pred == 1]
x2 = X[label_pred == 2]
```

客户类型聚类（3）

```
plt.scatter(x0[:,1], x0[:,0], c = "red", marker='o', label='label0')

plt.scatter(x1[:,1], x1[:,0], c = "green", marker='*', label='label1')

plt.scatter(x2[:,1], x2[:,0], c = "blue", marker='+', label='label2')

plt.xlabel('Age')

plt.ylabel('Income')

plt.legend()

plt.show()
```



机器学习步骤

一. 导入数据

二. 概述数据

三. 数据可视化

四. 评估算法

五. 实施预测

一. 导入数据

```
# 导入类库
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
```

一. 导入数据

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
# 导入数据
dataset = pd.read_csv("iris.csv")
```

二. 概述数据

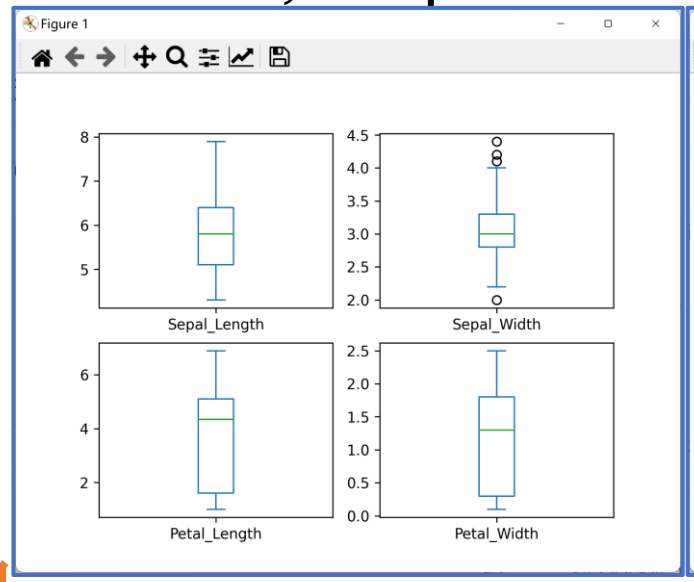
```
#显示数据维度
print('数据维度: 行 %s，列 %s' % dataset. shape)

# 查看数据的前10行
print(dataset. head(10))

# 统计描述数据信息
print(dataset. describe())

# 种类分布情况
print(dataset. groupby('Species').size())
```

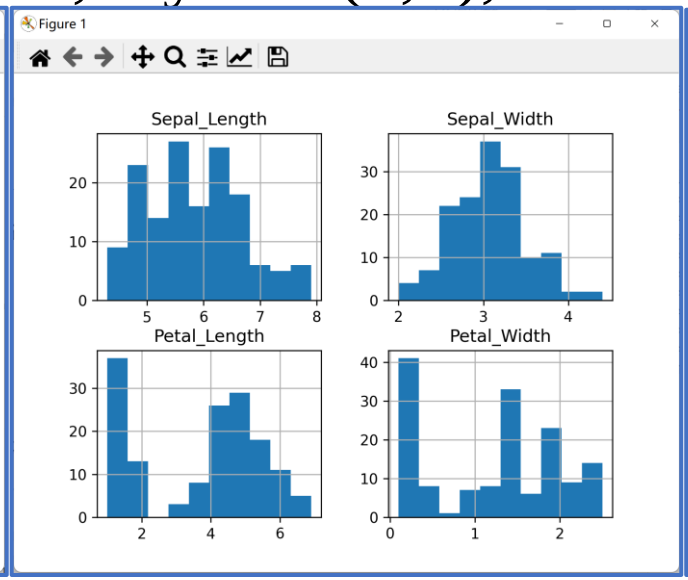
```
数据维度: 行 150，列 5
   Sepal_Length  Sepal_Width  Petal_Length  Petal_Width Species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
5           5.4          3.9           1.7          0.4  setosa
6           4.6          3.4           1.4          0.3  setosa
7           5.0          3.4           1.5          0.2  setosa
8           4.4          2.9           1.4          0.2  setosa
9           4.9          3.1           1.5          0.1  setosa
```

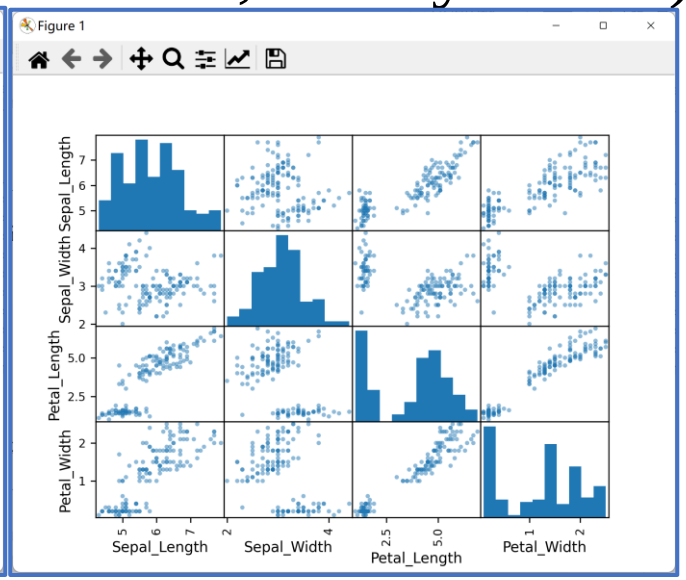
```
       Sepal_Length  Sepal_Width  Petal_Length  Petal_Width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
Species
setosa        50
versicolor    50
virginica     50
dtype: int64
```

三. 数据可视化

```
# 箱线图
dataset. plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False)
plt. show()

# 直方图
dataset. hist()
plt. show()

# 散点矩阵图
pd. plotting. scatter_matrix(dataset)
plt. show()
```







四. 评估算法

- 分离出评估数据集
- 采用10折交叉验证来评估算法模型
- 生成不同的模型来预测数据
- 选择最优模型

四. 评估算法（1）

```
# 分离数据集
array = dataset. values
X = array[:, 0:4]
Y = array[:, 4]
validation_size = 0.2  # 80%训练集，20%验证数据集
seed = 1 #随机数种子
X_train, X_validation, Y_train, Y_validation = \
    train_test_split(X, Y, test_size=validation_size, random_state=seed)
```

四. 评估算法（2）

```
# 算法审查
models = {}
models['LR'] = LogisticRegression(random_state=0, max_iter=1000)
models['KNN'] = KNeighborsClassifier()
models['CART'] = DecisionTreeClassifier()
models['NB'] = GaussianNB()
models['SVM'] = SVC()
```

四. 评估算法（3）

```
# 评估算法
results = []
for key in models:
    kfold = KFold(n_splits=10)
    cv_results = cross_val_score(models[key], X_train, Y_train,
cv=kfold, scoring='accuracy')
    results.append(cv_results)
    print('%s: %f (%f)' %(key, cv_results.mean(), cv_results.std()))
```

```
LR: 0.983333 (0.033333)
KNN: 0.983333 (0.033333)
CART: 0.975000 (0.038188)
NB: 0.975000 (0.053359)
SVM: 0.983333 (0.033333)
```

五. 实施预测

#使用评估数据集评估算法

```
svm = SVC()
svm.fit(X=X_train, y=Y_train)
predictions = svm.predict(X_validation)
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))
```

```
[[ 7  0  0]
 [ 0 10  2]
 [ 0  2  9]]
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00         7
  versicolor       0.83      0.83      0.83        12
   virginica       0.82      0.82      0.82        11

    accuracy                           0.87        30
   macro avg       0.88      0.88      0.88        30
weighted avg       0.87      0.87      0.87        30
```

评价指标

分类常用的评价指标：混淆矩阵 (Confusion Matrix) 、精确率 (Precision)、召回率 (Recall)、F1分数 (F1 Score)和准确率 (**Accuracy**)等。

回归主要评价指标：平均绝对误差(MAE，Mean absolute error)、均方误差(MSE，Mean squared error)、均方根误差(RMSE，Root Mean squared error)、R^2等。

混淆矩阵

混淆矩阵（误差矩阵）： 是表示精度评价的一种标准格式，用n行n列的矩阵形式来表示。

真实结果	预测结果	
	正例	反例
正例	(True Positive , TP) (预测为正，真实为正)	(False Negative, FN) (预测为负，真实为正)
反例	FP(False Positive , FP) (预测为正，真实为负)	(True Negative , TN) (预测为负，真实为负)

分类评价指标

精确率（Precision）又叫查准率：它是针对预测结果而言的，它的含义是在**所有被预测为正的样本中实际为正的样本比例**，其计算公式为：

$$P=\frac{TP}{TP+FP}$$

分类评价指标

召回率（Recall）又叫查全率：它是针对原样本而言的，它的含义是在**实际为正的样本中被预测为正样本的**比例，其公式如下：

$$R=\frac{TP}{TP+FN}$$

分类评价指标

F1分数（F1 Score）：是一个**综合精确率和召回率**的评价指标，当模型的精确率和召回率冲突时，可以采用该指标来衡量模型的优劣，其计算公式为：

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

分类评价指标

准确率（Accuracy）：是分类问题中最为常用的评价指标，准确率的定义是**预测正确的样本数占总样本数的比例**，其计算公式为：

$$ACC=\frac{TP+TN}{TP+TN+FP+FN}$$

案例分析

y_true	猫	猫	猫	猫	猫	狗	狗	狗	狗	狗
y_pred	猫	猫	猫	猫	狗	猫	狗	狗	猫	狗

其中：y_true代表样本的真实值，y_pred代表样本的模型预测值

假设：以猫为正，狗为负

结果：TP=4，FN=1，FP=2，TN=3。

案例分析

通过上述公式，可以计算出相关评价指标：

$$P=\frac{4}{4+2}\approx 0.67$$

$$R=\frac{4}{4+1}=0.8$$

$$F1=\frac{2\cdot P\cdot R}{P+R}\approx 0.73$$

$$ACC=\frac{4+3}{4+3+1+2}=0.7$$

案例分析

```
from sklearn import metrics    #评估指标
from sklearn.preprocessing import LabelBinarizer   #标签二值化
lb = LabelBinarizer()
y_true = ['猫', '猫', '猫', '猫', '猫', '狗', '狗', '狗', '狗', '狗']
y_pred = ['猫', '猫', '猫', '猫', '狗', '猫', '狗', '狗', '猫', '狗']
#  （1）计算混淆矩阵
print('Confusion Matrix：')
print(metrics. confusion_matrix(y_true, y_pred, labels=['猫', '狗']))
```

案例分析

```
# （2）将标签二值化，计算精确率、召回率、F1分数和准确率
y_true_binarized = lb. fit_transform(y_true)
y_pred_binarized = lb. fit_transform(y_pred)
print('精确率： %s' % metrics. precision_score(y_true_binarized, y_pred_binarized))
print('召回率： %s' % metrics. recall_score(y_true_binarized, y_pred_binarized))
print('F1分数：%s' % metrics. f1_score(y_true_binarized, y_pred_binarized))
print('准确率： %s' % metrics. accuracy_score(y_true_binarized, y_pred_binarized))
```

案例分析

```
# （3）classification_report()函数实现对精确率、召回率、F1分数和准确率的计算
print('Classification Report： ')
print(metrics.classification_report(y_true, y_pred))
```

```
Confusion Matrix:
[[4 1]
 [2 3]]
精确率： 0.6666666666666666
召回率： 0.8
F1分数: 0.7272727272727272
准确率： 0.7
Classification Report:
              precision    recall  f1-score   support

          狗       0.75      0.60      0.67         5
          猫       0.67      0.80      0.73         5

    accuracy                           0.70        10
   macro avg       0.71      0.70      0.70        10
weighted avg       0.71      0.70      0.70        10
```

评价指标

分类常用的评价指标：混淆矩阵 (Confusion Matrix) 、精确率 (Precision)、召回率 (Recall)、F1分数 (F1 Score)和准确率 (Accuracy)等。

回归主要评价指标：平均绝对误差(MAE，Mean absolute error)、均方误差(MSE，Mean squared error)、均方根误差(RMSE，Root Mean squared error)、R^2等。

案例分析

y_true	1	2	3
y_pred	2	3	4
y_pred2	1	3	5

y_true代表样本的真实值，**y_pred**代表该样本的模型预测值，**y_pred2**代表该样本的第二个模型的预测值。

回归评价指标

MAE（平均绝对误差）：对于回归模型性能评估最直观的思路是**利用模型的预测值与真实值的差值来衡量**，误差越小，回归模型的拟合程度就越好，其计算公式为：

$$\mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}\left|y_i-\hat{y}_i\right|$$

n 为样本的个数，y_i 为第 i 个样本的真实值，$\hat{y}_i$ 为第 i 个样本的模型预测值。

案例分析

通过上述公式以y_true和y_pred2为例计算出相关MAE：

y_true	1	2	3
y_pred	2	3	4
y_pred2	1	3	5

$$\mathrm{MAE}=\frac{|1-1|+|2-3|+|3-5|}{3}=1$$

回归评价指标

MSE（均方误差）：它是一种常用的回归损失函数，计算方法是**求误差的平方和**，由这两个指标的原理可知MSE比MAE对异常值更敏感，其计算公式为：

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$$

RMSE（均方根误差）：对均方误差进行开平方运算。

案例分析

通过上述公式以y_true和y_pred2为例计算出**MSE**：

y_true	1	2	3
y_pred	2	3	4
y_pred2	1	3	5

$$\mathrm{MSE}=\frac{(1-1)^2+(2-3)^2+(3-5)^2}{3}=\frac{5}{3}\approx 1.67$$

回归评价指标

决定系数R^2： 由MAE和MSE的公式可知，随着样本数量的增加，这两个指标也会随之增大，而且针对不同量纲的数据集，其计算结果也有差异，所以很难直接用这些评价指标来衡量模型的优劣，可以使用**决定系数R^2来评价回归模型的预测能力**。

回归评价指标

R^2计算公式为·

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

其中，$\bar{y}$ 表示 y 的均值。R^2 取值范围一般是 0～1，越接近 1，回归的拟合程度就越好。但当回归模型的拟合效果差于取平均值时的效果时，也可能为负数。

案例分析

通过上述公式以y_true和y_pred2为例计算R^2：

y_true	1	2	3
y_pred	2	3	4
y_pred2	1	3	5

$$\overline{y}=\frac{1+2+3}{3}=2$$

$$R^2=1-\frac{(1-1)^2+(2-3)^2+(3-5)^2}{(1-2)^2+(2-2)^2+(3-2)^2}=1-\frac{5}{2}=-1.5$$

案例分析

```
from sklearn import metrics
y_true = [1, 2, 3]
y_pred = [2, 3, 4]
y_pred2 = [1, 3, 5]
# （1）计算MAE
print('MAE：')
print('y_pred MAE：  %s' % metrics.mean_absolute_error(y_true, y_pred))
print('y_pred2 MAE：  %s' % metrics.mean_absolute_error(y_true, y_pred2))
```

案例分析

```
# （2）计算MSE
print('MSE：')
print('y_pred MSE： %s' % metrics.mean_squared_error(y_true, y_pred))
print('y_pred2 MSE：%s' % metrics.mean_squared_error(y_true, y_pred2))
# （3）计算决定系数
print('R2：')
print('y_pred R2： %s' % metrics.r2_score(y_true, y_pred))
print('y_pred2 R2：%s' % metrics.r2_score(y_true, y_pred2))
```

案例分析

```
MAE:
y_pred MAE:  1.0
y_pred2 MAE: 1.0
MSE:
y_pred MSE:  1.0
y_pred2 MSE: 1.6666666666666667
R2:
y_pred R2:  -0.5
y_pred2 R2: -1.5
>>>
```

Python应用领域

文本分析： Jieba、Nltk…

科学计算： Numpy、SciPy…

数据分析： Pandas、Matplotlib…

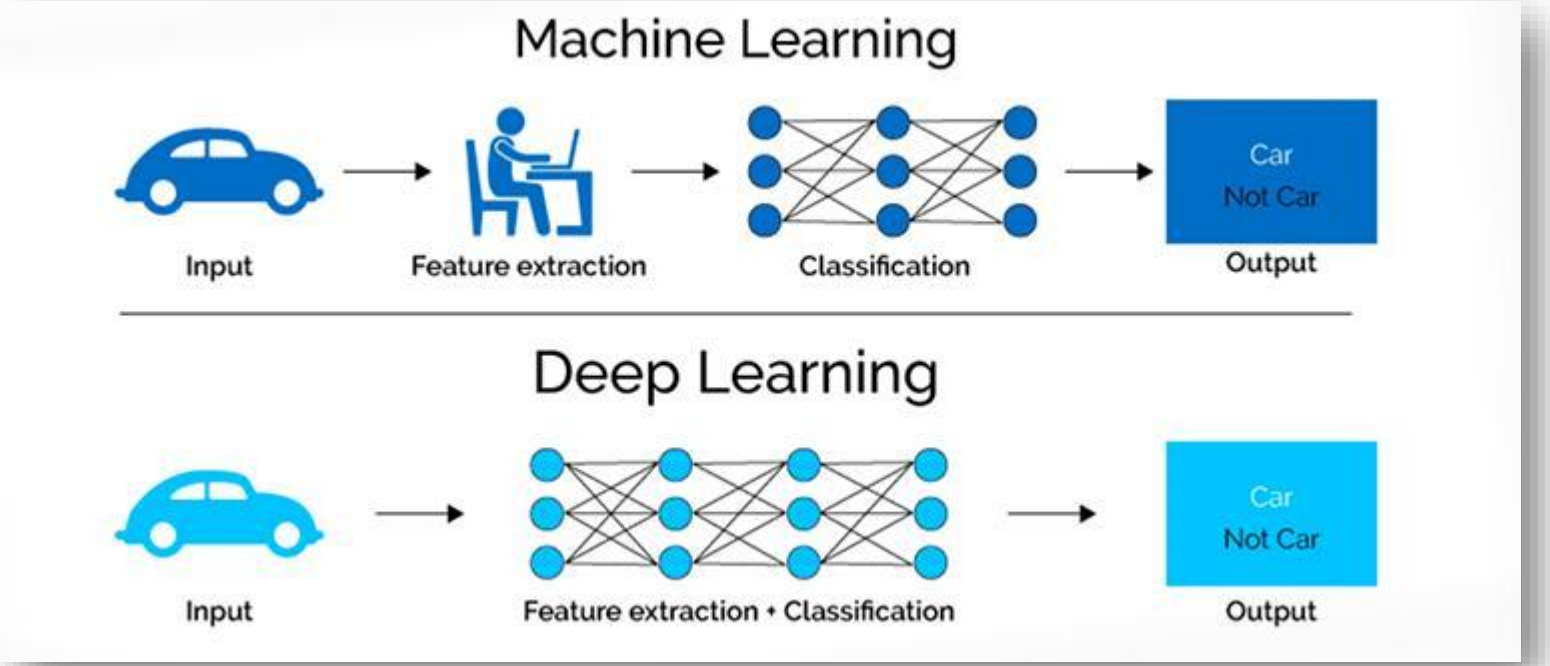
机器学习： Scikit-Learn、Keras…

深度学习： Pytorch、Mindspore、PaddlePaddle…

深度学习地位


人工智能
能够感知、推理、行动和适应的程序
机器学习
能够随着数据量的增加不断改进性能的算法
深度学习
机器学习的一个子集：利用多层神经网络从大量数据中进行学习

机器学习与深度学习


Machine Learning
Input
Feature extraction
Classification
Car
Not Car
Output
Deep Learning
Input
Feature extraction + Classification
Car
Not Car
Output

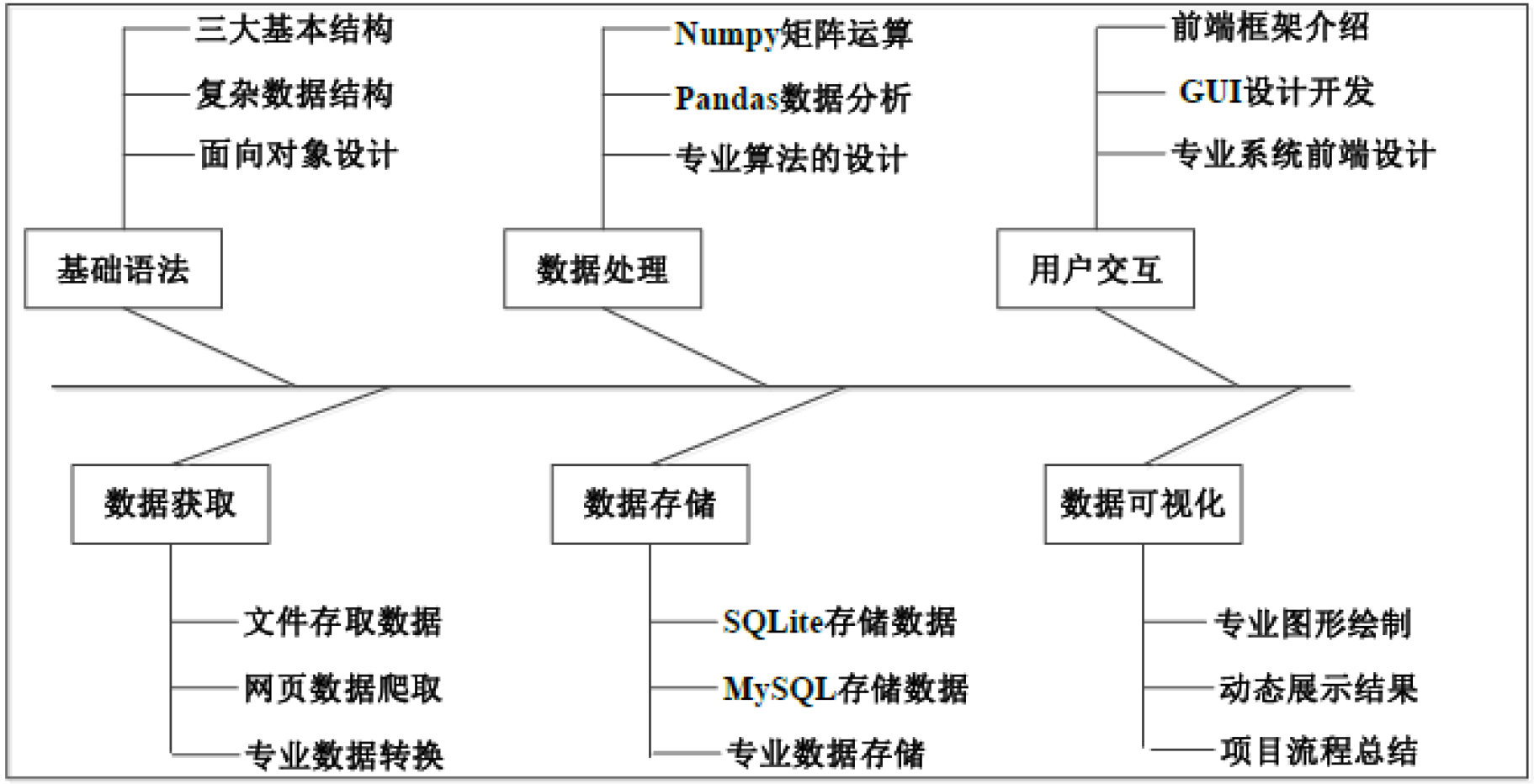
常用的深度学习框架

PyTorch： 由Facebook的团队开发，并于2017年在GitHub 上开源。

MindSpore： 由华为推出的新一代全场景AI计算框架，2020年MindSpore正式开源。

PaddlePaddle： 由百度推出的中国首个自主研发、功能丰富、开源开放的深度学习平台。

课程知识图谱


三大基本结构
复杂数据结构
面向对象设计
基础语法
Numpy矩阵运算
Pandas数据分析
专业算法的设计
数据处理
前端框架介绍
GUI设计开发
专业系统前端设计
用户交互
数据获取
文件存取数据
网页数据爬取
专业数据转换
数据存储
SQLite存储数据
MySQL存储数据
专业数据存储
数据可视化
专业图形绘制
动态展示结果
项目流程总结

题型及分值

考试题型如下：

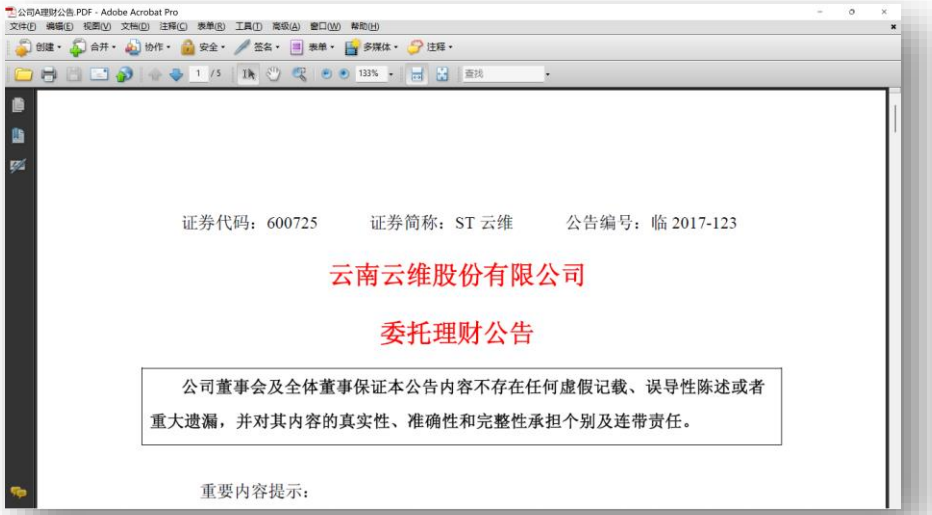
选择题：2分*20题=40分，涵盖全部教学内容

程序填空题：2分*3空*5题=30分，涵盖全部教学内容

编程题：10分*3题=30分

复习重点：窗体设计、matplotlib图形绘制(散点图、折线图、饼图、条形图等)、numpy数据处理、pandas数据分析、数据清洗、文献词频统计、正则与MySQL数据库、机器学习算法应用(分类、回归、聚类)

中英文献分析



证券代码：600725　　证券简称：ST 云维　　公告编号：临 2017-123

云南云维股份有限公司

委托理财公告

公司董事会及全体董事保证本公告内容不存在任何虚假记载、误导性陈述或者重大遗漏，并对其内容的真实性、准确性和完整性承担个别及连带责任。

重要内容提示：

AI in Finance: Challenges, Techniques, and Opportunities

LONGBING CAO, University of Technology Sydney, Australia

AI in finance refers to the applications of AI techniques in financial businesses. This area has attracted attention for decades, with both classic and modern AI techniques applied to increasingly broader areas of finance, economy, and society. In contrast to reviews on discussing the problems, aspects, and opportunities of finance benefited from specific or some new-generation AI and data science (AIDS) techniques or the progress of applying specific techniques to resolving certain financial problems, this review offers a comprehensive and dense landscape of the overwhelming challenges, techniques, and opportunities of AIDS research in finance over the past decades. The challenges of financial businesses and data are first outlined, followed by a comprehensive categorization and a dense overview of the decades of AIDS research in finance. We then structure and illustrate the data-driven analytics and learning of financial businesses and data. A comparison, criticism, and discussion of classic versus modern AIDS techniques for finance follows. Finally, the open issues and opportunities to address future AIDS-empowered finance and finance-motivated AIDS research are discussed.

网络爬虫

熟练掌握正则表达式的规则及应用…

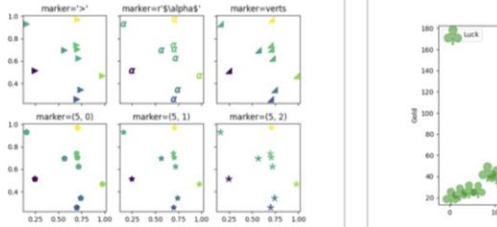
数据库设计

- SQLite
- MySQL
- MongoDB
- Redis
- Microsoft SQL Server 2000
- ….

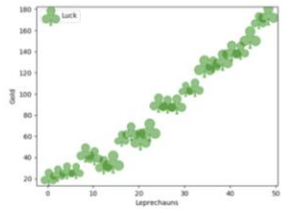
tkinter设计步骤

- 导入tkinter模块
- 创建GUI主窗体
- 添加人机交互控件并编写相应的函数
- 在主事件循环中等待用户触发事件响应

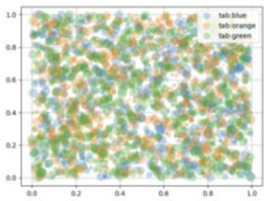
Matplotlib



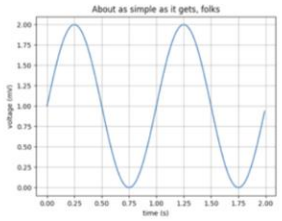
Marker examples



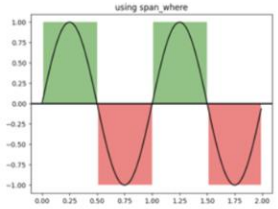
Scatter Symbol



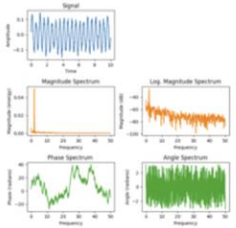
Scatter plots with a legend



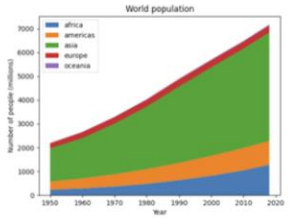
Simple Plot



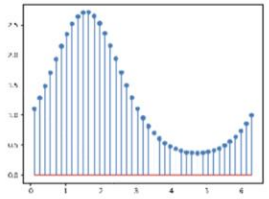
Using span_where



Spectrum Representations



Stackplots and streamgraphs



Stem Plot

Numpy

NumPy(Numerical Python的缩写)：是一个开源的Python科学计算库，NumPy数组在数值运算方面的效率优于列表。它是数据分析、机器学习和科学计算的主力军。

官网：**https://numpy.org/doc/stable/**

Pandas

Pandas ： 基于NumPy 的一种工具，该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的数据模型，提供了大量能快速便捷地处理数据的函数和方法。

Pandas有三个重要的数据结构：一维系列(Series)和二维数据框(DataFrame)、三维(Panel)。

官网： https://pandas.pydata.org/

scikit-learn

scikit-learn： 基于NumPy, SciPy, matplotlib，可以实现数据预处理、分类、回归、降维、聚类、模型选择等常用的机器学习算法，是数据挖掘和数据分析的一个简单有效工具。

机器学习分类：有监督学习、无监督学习

谢　谢