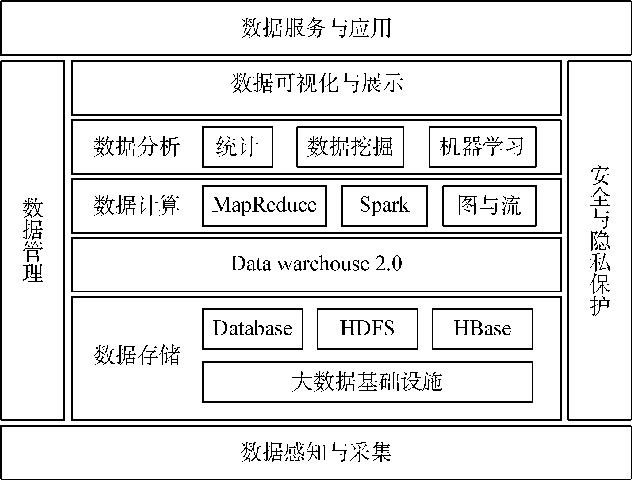
1数据的全生命周期管理包括哪些阶段 ?



如图所示,生命周期包括数据采集,数据储存,数据计算,数据分析,数据可视化展示,数据服务与应用

2数据采集的概念是什么 ? 都有哪些方法 ?

数据采集是指从真实世界对象中获得原始数据的过程

数据采集有三种方法,分别是,传感器,日志文件和web爬虫

3什么是数据管理 ? 比较传统的数据管理和大数据管理技术有什么异同 ?

数据管理是利用计算机硬件和软件技术对数据进行有效的收集,存储,处理和应用的过程

不同点:

传统数据管理主要处理结构化数据，如关系型数据库（SQL），然而大数据管理能够处理结构化、半结构化和非结构化数据（如文本、图像、视频）。

传统数据管理通常针对较小的、可管理的数据集，数据量在可接受范围内。然而大数据管理海量数据：处理大规模数据集，通常超过传统数据库的处理能力。

相同点：

数据存储：两者都需要有效的数据存储解决方案，以便组织和管理数据。

数据处理：无论是传统还是大数据管理，都涉及对数据的处理和分析，以提取有用的信息和洞察。

数据访问：都需要提供有效的数据访问和检索机制，确保用户能够方便地获取所需数据。

4大数据的计算模式可以分为哪几类 ?

数据的计算模式大致分为批量计算模式,流式计算模式,交互式计算模式和图计算模式四类.

5什么是数据分析 ? 有哪些数据分析的方法或者模型 ?

数据分析处理来自对某一兴趣现象的观察，测量或者实验的信息，数据分析目的，是从和主题相关的数据中提取尽可能多的信息，主要目标包括，推测或解释数据并确定如何使用数据，检查数据是否合法，给决策提供合理建议，诊断或推断错误

根据根据数据分析深度将数据分析分为三个层次，描述性分析，预测性分析和规则性分析，模型包括机器学习模型，线性回归，随机森林，支持向量机……

6数据可视化的原因有哪些 ?

数据可视化的原因主要包括以下几点：

 简化复杂数据：可视化可以将大量复杂的数据以图形或图表的形式呈现，使得用户更容易理解和分析。

 揭示模式和趋势：通过可视化，数据中的潜在模式、趋势和关系更容易被识别，有助于进行深入分析和决策。

地理解数据背后的含义。

 支持决策过程：清晰的可视化图表可以帮助管理层和决策者更快地掌握关键信息，从而做出更明智的决策.提高数据洞察力：可视化使得用户能够直观地查看数据，快速获得洞察，帮助他们更好

 促进沟通和分享：可视化可以有效地传达信息，使数据故事更具吸引力，便于在团队或组织内部分享和沟通。

练习题部分:

代码如下,图片在仓库

注意,应当plt.clf(),否则下一张图会有上一张图的信息

import matplotlib.pyplot as plt

import numpy as np

x = np.linspace(0, 10, 100)

y = np.sin(x)

plt.plot(x, y)

plt.title('Sine Wave')

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.savefig('linspace\_matplot.png')

plt.clf()

# plt.show()

x = np.random.rand(50)

y = np.random.rand(50)

plt.scatter(x, y)

plt.title('Random Scatter Plot')

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.savefig('Random Scatter\_matplot.png')

plt.clf()

# plt.show()

data = np.random.randn(1000)

plt.hist(data, bins=30, alpha=0.7)

plt.title('Histogram of Random Data')

plt.xlabel('Value')

plt.ylabel('Frequency')

plt.savefig('sine\_wave\_matplot.png')

plt.clf()

# plt.show()

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# Sine Wave

x = np.linspace(0, 10, 100)

y = np.sin(x)

sns.lineplot(x=x, y=y)

plt.title('Sine Wave')

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.savefig('sine\_wave\_seaborn.png')

plt.clf()

# plt.show()

# Random Scatter Plot

x = np.random.rand(50)

y = np.random.rand(50)

sns.scatterplot(x=x, y=y)

plt.title('Random Scatter Plot')

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.savefig('Scatter Plot\_seaborn.png')

plt.clf()

# plt.show()

# Histogram of Random Data

data = np.random.randn(1000)

sns.histplot(data, bins=30, kde=True, color='blue', alpha=0.7)

plt.title('Histogram of Random Data')

plt.xlabel('Value')

plt.ylabel('Frequency')

plt.savefig('histplot\_seaborn.png')

plt.clf()

# plt.show()