



CHAPTER 3

第 3 章 大模型编写程序

3.1 大模型实现编程功能

使用大模型编写代码是近年来人工智能和软件开发领域的一项创新技术，其核心在于将自然语言处理（NLP）技术应用于代码的创建，从而大幅提高软件开发效率，降低错误率。

开发人员可以描述他们需要的功能，AI 模型会自动将这些描述翻译成可用代码，而不再需要逐行编写代码。这些模型在包括代码库、技术论坛、编程平台、文档和相关网络数据的海量数据集上进行训练。这种广泛的训练使它们能够理解代码的上下文，包括注释、函数名称和变量名称，从而生成更具上下文准确性的代码。LLM 通过简化编码任务并减少从头编写代码所需的时间和精力，彻底改变了代码生成方式。它们有可能改变软件开发过程，使其对非技术用户更为友好。

目前，市场上已经有几种大模型代码生成产品，如 ChatGPT、AIXCoder、CodeGeex 和 GitHubCopilot，这些产品的功能特点如下表：

产品名称	功能
ChatGPT	可用于自然语言描述中的代码生成，开发人员可以提出问题或描述代码需求，ChatGPT 生成相关代码片段
AIXCoder	利用自然语言描述，为开发人员提供自动代码建议和完成代码段，与 IDE 紧密集成
CodeGeeX	提供自定义的代码生成工具，包括领域特定的代码生成工具，可根据客户的特定需求生成代码
GitHub Copilot	利用自然语言和上下文为开发人员提供代码建议，与 VS Code 等 IDE 紧密集成，为开发人员提供实时代码建议和自动生成代码片段

代码生成原则

- LLM 利用其在代码库、技术论坛和其他相关来源上的广泛训练来理解代码的上下文。这包括注释、函数名称和变量名称，使其能够生成更具上下文准确性的代码。
- LLM 使用自然语言处理技术来解析和解释开发人员提供的自然语言描述。这涉及将输入

分解成有意义的单元，理解描述中不同部分之间的关系，并将自然语言输入映射到适当的代码结构。

- LLM 通过利用其对编程语言和编码最佳实践的学习生成代码。这包括生成符合所选编程语言的语法和结构的代码，并纳入软件开发中常用的编码约定和模式。

例子

以下是一个利用 GPT-4o 生成静态网页的例子：

• 问题：

根据以下数据绘制图表，呈现在 html 中：

账 期 202201 202202 202203 202204 202205 202206 202207 202208 202209 202210 202211
202212 202301 202302 202303

携出用户 6250 6013 7311 8418 6109 8257 10388 11190 11154 10657 6932 6261 8852 13915
14822

携入移动 4871 4834 5692 6499 5065 6739 8266 8801 9146 8347 5239 5053 7257 11585 12568

携入电信 1379 1179 1619 1919 1044 1518 2122 2389 2008 2310 1693 1208 1595 2330 2254

=====

注意使用中文编码

• 回答：

要在 HTML 中绘制这些数据的图表，你可以使用 JavaScript 图表库来实现，比如 Chart.js 或者 Highcharts。这里我演示一下如何使用 Chart.js 来绘制一个简单的折线图。首先，你需要在 HTML 文件中引入 Chart.js 库：

```
<!DOCTYPE html>
<html lang="zh-CN">
<head>
<meta charset="UTF-8">
<title>携号转网用户统计</title>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
<canvas id="myChart" width="800" height="400"></canvas>
```

```
<script>
// 数据
var labels = ["202201", "202202", "202203", "202204", "202205", "202206", "
202207",
              "202208", "202209", "202210", "202211", "202212", "202301", "
202302", "202303"];
var dataXieChu = [6250, 6013, 7311, 8418, 6109, 8257, 10388, 11190, 11154,
10657, 6932, 6261, 8852, 13915, 14822];
var dataXieRuMobile = [4871, 4834, 5692, 6499, 5065, 6739, 8266, 8801, 9146,
8347, 5239, 5053, 7257, 11585, 12568];
var dataXieRuTelecom = [1379, 1179, 1619, 1919, 1044, 1518, 2122, 2389,
2008, 2310, 1693, 1208, 1595, 2330, 2254];

// 绘制图表
var ctx = document.getElementById('myChart').getContext('2d');
var myChart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: labels,
    datasets: [{
      label: '携出用户',
      data: dataXieChu,
      borderColor: 'rgb(255,99,132)',
      backgroundColor: 'rgba(255,99,132,0.2)',
      tension: 0.1
    }, {
      label: '携入移动',
      data: dataXieRuMobile,
      borderColor: 'rgb(54,162,235)',
      backgroundColor: 'rgba(54,162,235,0.2)',
      tension: 0.1
    }, {
      label: '携入电信',
      data: dataXieRuTelecom,
      borderColor: 'rgb(75,192,192)',
      backgroundColor: 'rgba(75,192,192,0.2)',
      tension: 0.1
    }
  ]
},
  options: {
    scales: {
      y: {
        beginAtZero: true
      }
    }
  }
});
```



```

    }
  }
});
</script>
</body>
</html>

```

这段代码创建了一个包含三条折线的折线图，分别代表携出用户、携入移动和携入电信的数据趋势。你可以将这段代码保存为 HTML 文件，然后在浏览器中打开，就可以看到生成的图表了。

下图为代码的运行结果：

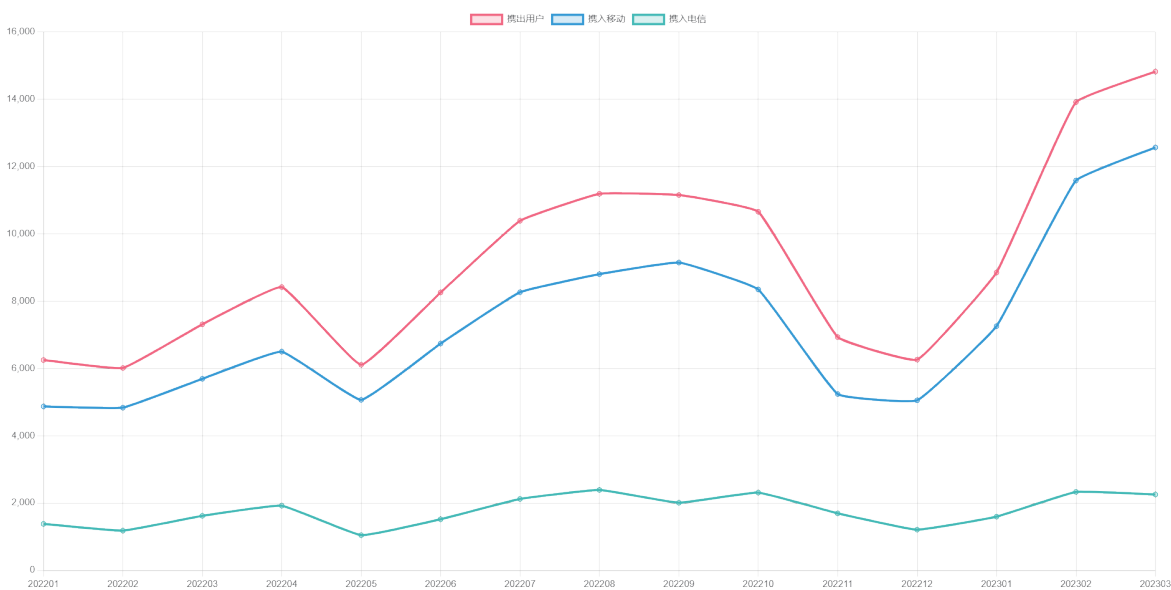


图 3.1: 由 ChatGPT-4o 生成的画图代码运行结果 (折线图)

• 问题：

帮我改成柱状图

• 回答：

好的，将图表类型从折线图改为柱状图非常简单，只需要将图表的类型从 'line' 改为 'bar' 即可。下面是修改后的代码：(略)

下图为更改后的代码运行结果：



图 3.2: 由 ChatGPT-4o 生成的画图代码运行结果 (柱状图)

编写 python 程序

ChatGPT-4o 能够编写各种编程语言的代码，其中对 python 的支持度最高。python 也是 GPT 唯一能够在服务器直接运行的语言。

复制某一文件

- 问题：

如何使用 python 复制某一文件。

- 回答：

要在 Python 中复制文件，你可以使用标准库中的 `shutil` 模块。这个模块提供了一个名为 `copyfile` 的函数，可以用来复制文件。下面是一个简单的示例，演示如何使用 `shutil.copyfile` 来复制一个文件：

```
import shutil

# 源文件路径
source_file = 'path/to/your/source/file.txt'

# 目标文件路径
target_file = 'path/to/your/target/file.txt'

# 使用 shutil.copyfile 进行文件复制
shutil.copyfile(source_file, target_file)

print(f'文件 {source_file} 已成功复制到 {target_file}')
```

请确保将 `source_file` 和 `target_file` 替换为实际的文件路径。这段代码将从 `source_file` 复制内容到 `target_file`。

编写深度学习模型项目

假设我需要做一个“根据多重因素预测室内植物灌溉”的研究工作，下面展示了利用 GPT 进行研究并编写程序的流程。

• 问题：

产生一些关于“根据气温气压光照等因素预测室内植物灌溉”的知识。

• 回答：

气温影响

– 植物的水分蒸发率与环境温度密切相关。温度较高时，植物蒸发水分较快，因此需要增加灌溉频率。

– 低温条件下，植物蒸发水分减少，灌溉频率可以降低。

气压影响

– 气压变化通常会影响植物的水分吸收速度。气压较高时，水分被土壤保持得较好，灌溉频率可以适当减少。

– 气压较低时，土壤的水分散失较快，因此需要增加灌溉频率。

光照影响

– 光照充足时，植物的光合作用活跃，水分消耗增加，灌溉频率应相应增加。

– 光照不足时，植物的生长减慢，水分需求降低，可以减少灌溉次数。

综合考虑

– 综合考虑气温、气压和光照等因素的变化，可以制定灌溉计划。例如，可以根据天气预报调整灌溉频率和量，确保植物在合适的水分条件下生长。

这些因素可以通过智能灌溉系统或者手动监测与调整来实现，以确保室内植物能够得到适当的水分供给，促进其健康生长。

• 问题：

除了我提到的光照、气压、温度，还有什么其他因素会影响室内植物的灌溉量。

• 回答：

湿度影响

– 空气湿度高会减少植物的蒸发速率，可能降低灌溉频率。

– 低湿度则可能需要增加灌溉次数，以补充更多的水分。

风速影响

- 风速高会增加植物表面的蒸发速率，因此可能需要增加灌溉量，以保持土壤湿润。

季节变化

- 季节变化会影响光照、温度和空气湿度，从而直接影响植物的生长和水分需求。例如，夏季通常需要比冬季更频繁地灌溉。

植物生长阶段

- 植物的生长阶段会影响其水分需求。例如，处于生长期的植物通常需要比休眠期更多的水分。

土壤类型和质地

- 不同类型和质地的土壤对水分的保持能力不同，这会影响灌溉的频率和量。

植物种类和大小

- 不同种类和大小的植物对水分的需求也不同，一些植物可能需要比其他植物更多的水。

• 问题：

我想通过室内光照、气压、温度、湿度，预测室内植物的灌溉量，比如我需要知道 20 度的时候具体浇多少水，25 度的时候具体浇多少水，灌溉量的标签值应该怎么获取。

• 回答：

- 数据收集与记录：

- * 确保有可靠的传感器或测量工具，用于实时监测室内的光照强度、气压、温度和湿度。
- * 将这些数据按照时间序列记录下来，以便后续分析和建模使用。

- 标签值的获取：

- * 标签值指的是你希望预测的目标，即灌溉量。这可以通过以下方法获取：

手动测量法：在实验阶段，可以手动记录每次灌溉的具体量，并与监测到的环境条件（光照、气压、温度、湿度）对应。

传感器测量法：安装灌溉系统的传感器，直接测量每次灌溉的水量，同时记录环境条件。

- 建立预测模型：

* 使用收集到的数据，可以采用机器学习或统计建模的方法来建立预测模型。例如，可以尝试以下方法：

回归模型：通过回归分析，建立光照、气压、温度、湿度与灌溉量之间的关系模型。

机器学习算法：使用支持向量机（SVM）、随机森林（Random Forest）或神经网络等算法进行预测。

– 模型评估与调整：

- * 训练模型后，使用未见过的数据进行测试和评估模型的预测能力。
- * 根据评估结果调整模型参数或选择更适合的算法，以提高预测的准确性和稳定性。

– 实时预测与调整：

* 部署模型后，可以实时输入当前的环境条件（光照、气压、温度、湿度），以预测室内植物所需的灌溉量。

- * 根据模型输出调整灌溉系统，以满足植物在不同环境条件下的水分需求。

• 问题：

现在需要根据环境的一些数据来预测室内植物的灌溉量，但是我目前还没有开始收集数据，你可以用 csv 的格式帮我产生 50 条数据吗，数据包括时间、光照、气压、温度、湿度、表头标注单位，我需要预测的结果是灌溉量。注意尽量模拟真实的数据变化和关系。

• 回答：

```
import pandas as pd
import numpy as np

# 生成样本数据
np.random.seed(0)

# 生成时间序列
time_series = pd.date_range(start='2024-01-01', periods=50, freq='H')

# 生成每个变量的实际范围
light = np.random.uniform(100, 1000, size=50) # 光照强度 (单位: lux)
pressure = np.random.uniform(950, 1050, size=50) # 气压 (单位: hPa)
temperature = np.random.uniform(15, 35, size=50) # 温度 (单位: 摄氏度)
humidity = np.random.uniform(30, 90, size=50) # 湿度 (单位: 百分比)

# 根据环境因素模拟灌溉量
# 为简单起见，假设灌溉量是环境变量的函数
irrigation_amount = 0.1 * light + 0.05 * (1000 - pressure) + 0.2 * temperature - 0.1 *
                    humidity + np.random.normal(0, 5, size=50)

# 创建DataFrame
data = pd.DataFrame({
    'Time': time_series,
    'Light (lux)': light,
    'Pressure (hPa)': pressure,
    'Temperature (Celsius)': temperature,
    'Humidity (%)': humidity,
    'Irrigation Amount (ml)': irrigation_amount
})
```

```
# 保存为CSV文件
csv_path = '/mnt/data/sample_irrigation_data.csv'
data.to_csv(csv_path, index=False)

csv_path
```

- 问题：

我希望利用以上数据来预测灌溉量，我应该使用什么模型来完成？

- 回答：

根据你提供的数据特征，你可以考虑使用以下几种模型来预测灌溉量：

1. 线性回归模型：适合处理连续的数值型数据，可以通过拟合线性关系来预测灌溉量。
2. 决策树模型：可以处理非线性关系，并能够处理特征之间的交互效应。
3. 随机森林模型：是基于多棵决策树的集成学习模型，可以提供更准确的预测结果，同时能够处理特征的重要性评估。

4. 支持向量机（SVM）：适合处理复杂的非线性关系，通过核函数来将输入特征映射到高维空间进行分类或回归。

5. 神经网络模型：可以捕捉复杂的非线性关系，适合处理大量数据和复杂模式。

根据你的需求和数据特征，你可以先尝试使用简单的线性回归模型进行建模和预测。如果线性模型效果不佳或者需要更精确的预测，可以考虑使用决策树、随机森林等更复杂的模型。

- 问题：

我收集了一些实验数据，数据保存在 data.csv 的文件中，下面是数据的例子：

时间, 光照 (lux), 气压 (hPa), 温度 (摄氏度), 湿度 (%), 灌溉量 (ml)

2023-06-09 08:00,2500,1012,22.5,55,200

2023-06-09 09:00,2800,1014,23.2,52,220

2023-06-09 10:00,3200,1011,24.1,50,240

2023-06-09 11:00,3100,1010,24.5,48,230

2023-06-09 12:00,3400,1009,25.0,47,250

2023-06-09 13:00,3700,1012,25.5,46,270

我希望可以构建一个 LSTM 的深度学习的模型，根据光照、气压、温度、湿度这些特征来预测灌溉量。可以帮我写一个完整的 python 程序吗？包括数据处理的流程，以及模型搭建，模型训练的流程，和最后的模型测试流程。

- 回答：

下面是一个完整的 Python 程序，包含数据处理、LSTM 模型搭建、模型训练和模型测试流程。我们将使用 Keras 和 Pandas 库来实现这一过程。

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
import matplotlib.pyplot as plt

# 1. 读取数据
data = pd.read_csv('data.csv')

# 2. 数据预处理
# 转换时间列为 datetime 类型并设置为索引
data['时间'] = pd.to_datetime(data['时间'])
data.set_index('时间', inplace=True)

# 选择特征和目标变量
features = data[['光照(lux)', '气压(hPa)', '温度(摄氏度)', '湿度(%)']].values
target = data['灌溉量(ml)'].values

# 数据归一化
scaler = MinMaxScaler()
features_scaled = scaler.fit_transform(features)

# 3. 创建时间序列数据
def create_dataset(features, target, time_steps=1):
    X, y = [], []
    for i in range(len(features) - time_steps):
        X.append(features[i:(i + time_steps), :])
        y.append(target[i + time_steps])
    return np.array(X), np.array(y)

time_steps = 3 # 你可以调整这个参数
X, y = create_dataset(features_scaled, target, time_steps)

# 分割训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 4. 构建 LSTM 模型
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(time_steps, X.shape[2])))
model.add(Dropout(0.2))
model.add(LSTM(50, return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(1))

model.compile(optimizer='adam', loss='mean_squared_error')
```

```
# 5. 训练模型
history = model.fit(X_train, y_train, epochs=100, batch_size=32, validation_data=(X_test,
                                                                              y_test))

# 6. 模型测试
y_pred = model.predict(X_test)

# 可视化训练过程中的损失
plt.plot(history.history['loss'], label='train_loss')
plt.plot(history.history['val_loss'], label='val_loss')
plt.legend()
plt.show()

# 可视化预测结果
plt.figure(figsize=(10, 6))
plt.plot(y_test, label='True Value')
plt.plot(y_pred, label='Predicted Value')
plt.legend()
plt.show()
```

1. 读取数据：使用 `pandas` 读取 CSV 文件，并将时间列转换为 `datetime` 类型。
2. 数据预处理：选择特征和目标变量，对特征进行归一化处理。
3. 创建时间序列数据：定义一个函数 `create_dataset`，将数据转换为适合 LSTM 模型输入的时间序列格式。
4. 构建 LSTM 模型：使用 `Sequential` 搭建一个包含 LSTM 层和全连接层的神经网络模型。
5. 训练模型：使用训练集数据训练模型，并使用验证集数据监控模型的训练过程。
6. 模型测试：对测试集数据进行预测，并可视化真实值和预测值。

3.2 大模型解释代码

大语言模型本质上是一台可以理解和生成文本的机器，取决于其训练方式，有时大语言模型可以生成文本以外的内容。ChatGPT 就是这样一个大语言模型。我们可以像上一节那样利用大模型编写程序，实现某种功能。事实上，也可以反过来，即向 GPT 提供一个代码，并要求它做出解释，尤其是在面对代码量大、逻辑复杂和特定领域专业知识的代码，手动解释和理解往往耗时且困难。而大模型，如 GPT 凭借其强大的自然语言处理能力，能够在一定程度上帮助我们快速理解和解释代码。本节将探讨如何利用大模型来解释代码，以提高编程工作的效率和准确性。

使用大模型对代码进行解释，包括以下几个步骤：

- **了解大模型**：学习大模型的基本功能和工作原理，包括它如何生成和解释文本，适用于哪些编程任务。熟悉大模型所提供的各种功能，如代码解释、错误排查、代码优化等。通过阅读相关文档、教程和实践案例，掌握使用大模型的基础知识和技巧。

- **明确需求**：明确需要大模型帮助解释的代码类型（例如 Python、R、JavaScript 等）和代码的功能（例如算法、数据处理、网络请求等）。确定希望大模型提供的解释细节，例如函数作用、算法步骤、逻辑流程等。这一步有助于提升大模型生成解释的准确性和相关性。

- **选择合适的平台**：选择适合的平台来与大模型进行互动，例如 OpenAI 的官方网站、集成 GPT 的开发工具，或其他支持 GPT 的应用程序。不同平台可能提供不同的功能和接口，根据需求选择合适的工具，以便更高效地利用大模型进行代码解释。

- **与 GPT 交互**：向 GPT 提供代码片段，并清晰地描述希望得到的解释内容，例如代码的功能、算法原理或逻辑步骤。使用明确的提示语句，如：“请解释以下代码的功能和工作原理，以代码解释 + 代码片段的形式”。这样可以引导大模型生成更符合需求的解释。

- **提供详细的信息**：提供代码的背景信息和相关问题，帮助 GPT 更准确地理解需求。例如，“这是一个计算线性回归模型的代码，我希望了解每一部分的作用和实现方法”。详细的信息可以让大模型更好地理解代码的上下文，从而生成更准确的解释。

- **分析回复**：仔细阅读 GPT 的解释，确保自己理解了代码的功能、逻辑和实现细节。如果有不明白的地方，可以继续提问以获得进一步的解释或澄清。通过逐步理解和分析回复内容，逐渐掌握代码的整体结构和具体实现。仔细阅读 GPT 的解释，确保自己理解了代码的功能、逻辑和实现细节。如果有不明白的地方，可以继续提问以获得进一步的解释或澄清。

- **测试与调试**：根据 GPT 的解释进行测试和调试，验证解释的准确性。如果 GPT 的解释存在错误或不完整，可以继续调整和优化代码，或再次向 GPT 提问以获取更精确的解释。在实际操作中，通过不断实践和反馈，提升对代码的理解和掌握能力。

简单代码示例

在使用 GPT 解释代码之前，我们需要记住两个提示词用法：

- (1) 保证自己的问题明确且具体
- (2) 尝试在问题最后加上“请进一步思考”或“请认真回答”等语句。这种方法有助于提高 GPT 的回答精确度，只需要在提示词后加上这些提醒，ChatGPT 的回答就会有显著的不同。

下面，让我们考虑一个使用 GPT 解释代码的简单示例：

```

n = int(input('输入n: '))
original_n = n # 保存原始的n值
s = 0
while n >= 1:
    s += n
    n -= 1
print('1-%d 求和结果: %d' % (original_n, s)) # 使用original_n来
显示用户输入的n值

请帮我详细梳理出代码的主要流程，用小标题+代码解释/作用的形式
给出。

```

< 2/2 >



用户输入

python

复制代码

```
n = int(input('输入n: '))
```

- 提示用户输入一个整数n，并将其转换为整数类型。用户输入的值将作为计算1到n累加和的起始值。

保存原始的n值

python

复制代码

```
original_n = n
```

- 保存用户输入的原始n值，以便后续在打印结果时使用。这是为了在循环中改变n的值后，仍然能够显示用户输入的原始n值。

初始化累加和变量

python

复制代码

```
s = 0
```

- 初始化变量s，用于存储1到n的累加和。初始值为0。

计算累加和

python

复制代码

```
while n >= 1:
    s += n
    n -= 1
```

- 使用while循环，当n大于等于1时，继续执行循环体。
 - `s += n`：将当前的n值累加到s中。
 - `n -= 1`：将n的值减1，以便在下次循环中使用，直到n小于1时停止循环。

打印结果

python

复制代码

```
print('1-%d 求和结果: %d' % (original_n, s))
```

- 使用格式化字符串，显示用户输入的原始n值以及1到n的累加和。`original_n`用于显示用户最初输入的n值，而`s`用于显示计算得到的累加和结果。

🔊 📄 🔄 🔍 ↕

图 3.3：代码解释的简单示例

这是一个实现 1 到给定正整数 n 累加求和的简单示例。在示例中，关键提示词为“流程”。虽然提示语句比较简单，但我們需要注意如何向 GPT 提供代码：提示语句应在代码后边或前边，这有助于向 GPT 确定哪些是自己的指令，哪些不是。在提示语句中，采用了“小标题 + 代码解释 + 代码片段”的形式。我们也可以采用其它自己想要的形式，比如直接在代码后注释。

```
n = int(input('输入n: '))
original_n = n # 保存原始的n值
s = 0
while n >= 1:
    s += n
    n -= 1
print('1-{} 求和结果: {}'.format(original_n, s)) # 使用original_n来显示用户输入的n值
```

请帮我逐行解释代码的含义，直接在代码后注释即可

< 5/5 >



```
python 复制代码

n = int(input('输入n: ')) # 提示用户输入一个整数n，并将其转换为整数类型
original_n = n # 保存原始的n值，以便后续打印时使用
s = 0 # 初始化变量s，用于存储累加和
while n >= 1: # 当n大于等于1时，继续执行循环
    s += n # 将当前的n值累加到s中
    n -= 1 # 将n的值减1，以便在下次循环中使用
print('1-{} 求和结果: {}'.format(original_n, s)) # 使用original_n来显示用户输入的n值，并打印1到n
```

图 3.4: 代码解释的简单示例

这样，GPT 将直接在代码片段中给出注释。

代码改进以及解释

在上一个简单代码示例，我们可以看出，向 GPT 给出代码并指定解释方式可以帮助我们更快且容易地理解代码。有时，我们想让 GPT 在给定代码的基础上对代码加以改进优化。比如在实际编程中，我们经常遇到多种编程语言的转换问题，而 GPT 是一个可以使用多种编程语言的大模型，所以我们可以让 GPT 对代码的编程语言进行转换并作出解释。

```
hess = z.*repmat(wts2,1,p); % matrix sqrt of hessian
```

```
[hu,hs,hv] = svd(hess,0); 在R中怎么计算SVD
```



在R中，你可以使用`svd()`函数来计算奇异值分解（Singular Value Decomposition, SVD）。以下是在R中执行类似操作的代码示例：

```

R 复制代码

# 假设 z、wts2 是你的矩阵
# 计算 hess 矩阵
hess <- z * matrix(wts2, nrow = nrow(z), ncol = ncol(z), byrow = TRUE)

# 计算 SVD
svd_result <- svd(hess)

# 获取奇异值、左奇异向量和右奇异向量
hu <- svd_result$u
hs <- diag(svd_result$d)
hv <- svd_result$v

# 在R中，svd()函数返回一个列表，包含u、d（奇异值）和v。
# 你可以根据需要使用这些值进行后续计算。

```

请注意，这里的`z`和`wts2`是根据你提供的MATLAB代码中的`z`和`repmat(wts2, 1, p)`的相应部分进行的猜测。确保根据你的实际情况适当调整代码。

图 3.5: 代码改进的示例

在这个示例中，我们想解决一个 MATLAB 代码转换为 R 代码的问题，在给出指令语句后，GPT 给出了 hess 矩阵在 R 中的计算以及进行奇异值分解的代码还有解释。在 MATLAB 代码的第一行是将列向量 wts 横向复制 p 次生成矩阵 hess，对应的 R 代码使用 matrix 函数生成矩阵 hess。最后，R 中使用 svd 函数进行奇异值分解。

此外，当代码中结果与预期不相符合时我们还可以对给出的代码进行提问。

代码涉及到函数的默认参数问题，由于在 Python 中默认参数只在函数定义时被初始化一次，所以最后打印出的列表中有 1 和 2 两个元素。向 GPT 提问，可以看到 GPT 对于提问进行了详细回答并给出解决方法。

```

def append_to_list(value, list=[]):
    list.append(value)
    return list

print(append_to_list(1))
print(append_to_list(2))  为什么最后的结果不是[2]

```

< 2/2 >



这是因为你在定义函数时使用了一个可变对象（`list=[]`）作为默认参数。让我们详细解释一下这个行为。

在 Python 中，默认参数只在函数定义时被初始化一次。如果你使用的是一个可变对象（比如列表、字典或集合）作为默认参数，所有对该函数的调用都会共享这个对象。

来看你的代码：

```
python 复制代码

def append_to_list(value, list=[]):
    list.append(value)
    return list

print(append_to_list(1)) # 输出: [1]
print(append_to_list(2)) # 输出: [1, 2]
```

解释

1. 第一次调用 `append_to_list(1)` 时, `list` 是一个空列表 `[]`。`1` 被添加到这个列表中, 所以返回的列表是 `[1]`。
2. 第二次调用 `append_to_list(2)` 时, `list` 是之前调用时使用的那个列表, 而不是一个新的列表。因此, `2` 被添加到这个已经有 `1` 的列表中, 所以返回的列表变成了 `[1, 2]`。

解决方法

如果你希望每次调用函数时都使用一个新的列表, 可以将 `list` 参数的默认值设为 `None`, 然后在函数内部创建一个新的列表:

```
python 复制代码

def append_to_list(value, list=None):
    if list is None:
        list = []
    list.append(value)
    return list

print(append_to_list(1)) # 输出: [1]
print(append_to_list(2)) # 输出: [2]
```

在这个修正后的版本中, 每次你调用 `append_to_list` 时, `list` 参数都会被初始化为一个新的空列表, 避免了共享同一个列表的问题。

🔊 📄 🔄 🔍 ↻

图 3.6: 代码改进的示例

在一些特定领域的问题中, 我们往往需要在原有代码基础上对其中的关键部分进行修改。比如在时间序列分析中, 我们要生成指定阶数与参数值的 ARMA 模型, 可以在给定的生成 ARMA 模型的代码基础上让 GPT 进行修改。

```

m <- 50
n0 <- N+m
epsilon <- rnorm(n0,0,sigma2)
y0 <- 0
y1 <- 0
y <- c(y1)
for (i in 2:n0){

  if (i==2){
    y[i] <- 0.6*y[i-1] - 0.05*y0 + epsilon[i] + 0.4*epsilon[i-1]
  }else if (i>2){
    y[i] <- 0.6*y[i-1] - 0.05*y[i-2] + epsilon[i] + 0.4*epsilon[i-1]
  }

}

y <- y[(m+1):n0]

```

我想修改以下代码，生成ARMA (1,1) 模型，其中 ar1=0.6,ma1=0.4，仅给出修改部分的代码，未修改的代码不用展示

< 10/10 >

要生成一个 ARMA(1,1) 模型，您需要将 `y0` 和 `y1` 的初始化逻辑修改为适应 ARMA(1,1) 的结构，并且在生成 `y` 的过程中应用 ARMA(1,1) 的方程。以下是需要修改的部分代码：

```

r
复制代码

y <- numeric(n0) # Initialize y with zeros

for (i in 2:n0) {
  y[i] <- 0.6*y[i-1] + epsilon[i] + 0.4*epsilon[i-1]
}

y <- y[(m+1):n0]

```

图 3.7：代码改进的示例

我们将生成 ARMA(2,1) 模型的代码提供给 GPT，并指定了要新生成的 ARMA 模型的阶数以及对应的参数值，GPT 给出了生成新模型的代码。将 GPT 给出的代码与自己提供给它的代码相比，可以看出改变的代码只有向量 y 的生成方式。这是因为新生成的 ARMA 模型与原来的模型的区别只是当前序列值受过去几期序列值的影响。

代码编写以及解释

我们也可以像上一节那样，对大模型提出一个编写程序实现某种功能的要求并让大模型对编写的代码进行解释。

比如我们想找出一个矩阵中每一列的除零以外最小值，但一般的编程语言中只有寻找最小值的函数，因此需要自己编写函数来实现，我们可以借助大模型来完成函数的编写。

在R中，我想查看矩阵中每列 除了0之外的最小数，
`apply(new_baseh,2,min)`

在R中，您可以使用 `apply` 函数来查看矩阵中每列除0之外的最小数。您需要定义一个自定义函数来忽略0，并将其应用到矩阵的每一列。可以通过以下步骤实现：

1. 定义一个自定义函数，用于计算除0之外的最小数。
2. 使用 `apply` 函数，将自定义函数应用到矩阵的每一列。

下面是具体实现方法：

```
R 复制代码
# 示例矩阵
new_baseh <- matrix(c(0, 3, 1, 4, 0, 2, 0, 5, 6), nrow=3, byrow=TRUE)

# 定义一个自定义函数，返回除0之外的最小值
min_non_zero <- function(x) {
  non_zero_values <- x[x != 0]
  if (length(non_zero_values) == 0) {
    return(NA) # 如果全是0，返回NA
  } else {
    return(min(non_zero_values))
  }
}

# 使用apply函数将自定义函数应用到矩阵的每一列
min_values_per_column <- apply(new_baseh, 2, min_non_zero)

# 打印结果
print(min_values_per_column)
```

上述代码中，`min_non_zero` 函数首先过滤掉向量中的0，然后计算剩余值的最小值。如果某一列全是0，函数返回 `NA`。接着，通过 `apply` 函数将这个自定义函数应用到矩阵的每一列（即 `apply(new_baseh, 2, min_non_zero)`），从而得到每列除0之外的最小值。

图 3.8：代码编写的示例

由于在提示语句中，没有指定矩阵的具体形式，因此，GPT 给出了一个三行三列的示例矩阵。最后，GPT 给出了编写的函数代码并做出了解释。由于给出的函数代码比较繁琐，我们可以进一步要求 GPT 以较为简化的方式重写函数。

可以用1行代码实现 找出矩阵中每列除0之外最小值吗

当然可以，您可以通过嵌套使用 `apply` 和 `min` 函数，再加上 `subset` 来实现一行代码完成这个任务。以下是实现方法：

```
R 复制代码
min_values_per_column <- apply(new_baseh, 2, function(x) min(x[x != 0], na.rm = TRUE))
```


这一行代码中, ``apply(new_baseh, 2, function(x) min(x[x != 0], na.rm = TRUE))`` 对矩阵的每一列应用匿名函数, 该匿名函数会过滤掉列中的0值, 然后计算非0值的最小值。如果某一列全是0, ``min`` 函数会返回 ``Inf``。

图 3.9: 代码编写的示例

在给出新的提示语句后, GPT 这次仅给出了一行代码, 使用匿名函数的方式, 完成了相同的要求。此外, 我们可以要求更多, 例如要求捕获代码中的错误或安全问题, 让我们让 GPT 生成一些单元测试代码, 将提示语句“你能给我几个单元测试用例来确保这个函数运行良好吗?”输入给 GPT。

当然! 下面是几个单元测试用例来确保自定义函数 ``min_non_zero`` 和 ``apply`` 函数的正确性。

测试案例 1: 基本矩阵

输入矩阵:

```
R 复制代码
new_baseh <- matrix(c(0, 3, 1, 4, 0, 2, 0, 5, 6), nrow=3, byrow=TRUE)
```

预期输出:

```
R 复制代码
c(1, 3, 2)
```

测试案例 2: 包含零元素的矩阵

输入矩阵:

```
R 复制代码
new_baseh <- matrix(c(0, 0, 0, 1, 0, 2, 3, 4, 0), nrow=3, byrow=TRUE)
```

预期输出:

```
R 复制代码
c(1, 4, 2)
```

测试案例 3: 包含全部零元素的列

图 3.10: 测试用例

图中展示了 GPT 给出的部分测试用例, 这些测试用例涵盖了包含零元素与不包含零元素的三行三列矩阵。通过在编程软件中运行 GPT 给出的代码, 并将输出结果与预期输出进行比较, 可以验证函数是否正常工作。

```

随访1流水号_x HM_volume.1 随访2流水号_x HM_volume.2 随访3
流水号_x HM_volume.3 随访4流水号_x HM_volume.4 随访5流水
号_x
2.01612E+13 74902 2.01612E+13 70952 2.01612E+13
62831 2.01612E+13 44029
2.01604E+13 52271 2.01604E+13 47748 2.01604E+13
13055 2.01605E+13 20
2.01604E+13 106042 2.01604E+13 103263
2.01612E+13 39877 2.01612E+13 16622 2.01612E+13 8441
2.01612E+13 24472 2.01612E+13 25477
2.01611E+13 148092 2.01611E+13 74087 2.01611E+13
61875 2.01611E+13 18546
2.01612E+13 28715 2.01612E+13 27343 2.01612E+13
20521 2.01612E+13 366 2.01612E+13
2.01612E+13 33498 2.01612E+13 36859 2.01612E+13
30298
2.01611E+13 56613 2.01611E+13 53913

```

这是我导入PYTHON的数据，我想保留每一行中出现空缺值之前的最近的两个数，并生成新的两列来存放，请仅给出导入数据后的代码，其它的代码省略

图 3.11：提示语句

Python 在处理数据方面与其它编程软件相比具有明显优势，在我们处理 Excel 中的数据时可以借助大模型进行。由于所处理的数据的行数可能较大，所以我们可以将 Excel 中数据中前面的若干行复制并输入到 GPT 中，如上所示。这样我们就避免了将整个数据输入到 GPT，相反，只需要前几行以及所分析的几列输入 GPT 即能完成需求。这里，我们需要分析一种特殊的数据，需求是保留每一行中离空缺值最近的前两个值并生成新的两列去保存这两个数。在输入部分数据，指定编程语言和要求后，GPT 给出相应的代码并做出解释。

你可以使用以下代码来完成这个任务：

```

python 复制代码

import pandas as pd

# 假设你的数据已被导入到一个DataFrame中，命名为df
# df = pd.read_csv('your_data.csv') # 根据实际情况导入数据

# 定义一个函数来获取最近的两个非空值
def get_last_two_values(row):
    values = row.dropna().values # 获取非空值
    return pd.Series(values[-2:]) # 取最后两个非空值

```

```
# 对每一行应用该函数并生成新的两列
df[['Value1', 'Value2']] = df.apply(lambda row: get_last_two_values(row[1:]), axis=1) # 从
```

在这个代码中，我们通过定义一个函数来从每一行中提取最近的两个非空值，并将其存放到新的两列中。注意 `row[1:]` 用来忽略第一列流水号，你可以根据实际情况调整这一部分。

图 3.12: 代码编写与解释

记忆检测

在我们使用 GPT 进行代码解释时，可以对给定的代码或由 GPT 生成的代码要求进一步改进和优化。但要注意，由于 ChatGPT 的记忆有窗口限制，因此，在交流一段时间之后，如果还需要对原始的代码进行继续提问，那么可以使用提示语句来检测最初的那个问题它是否还记得。可以向 GPT 输入提示语句：“为了怕你忘记，检查一下，你还记得咱们刚刚讨论的是哪个代码吗，请给出代码名称或者函数名称的原始实现”。如果不记得，就需要自己重写输入代码，保证它的回答不会跑偏。

为了怕你忘记，检查你一下，你还记得咱们刚刚讨论的是哪个代码吗，请给出[代码名称或者函数名称]的原始实现。

当然记得，我们刚刚讨论的是一个名为 `range_x` 的函数。以下是该函数的原始实现：

```
python 复制代码
def range_x(start, stop, step):
    res = []
    while start < stop:
        res.append(start)
        start += step
    return res
```

图 3.13: 记忆检测

在输入提示语句后，GPT 给出了之前讨论的代码实现。接下来，我们就可以继续让 GPT 对代码进行解释和改进。

复杂代码解释

有时候对于庞杂的代码，或者复杂的 SQL 查询，我们需要梳理出核心线索。比如接下来的 SQL 查询代码中，表过多，比较乱，我们可以向 GPT 给出提示语句：“请帮我梳理出一条

最核心的线索，理解这段 SQL 查询”来帮助我们理解查询代码的逻辑。

核心线索

这条查询的核心目的是统计每个设备在“复旦大学”这个大学的用户在8月份的问题练习情况，包括总问题数和正确回答的问题数。

逐步解释

1. 选择字段：

```
sql 复制代码
select up.device_id, '复旦大学' as university,
       count(question_id) as question_cnt,
       sum(if(qpd.result='right', 1, 0)) as right_question_cnt
```

- `up.device_id`：从 `user_profile` 表中选择设备 ID。
- `'复旦大学' as university`：给结果中加入一列，值为“复旦大学”。
- `count(question_id) as question_cnt`：计算每个设备在8月份参与的问题练习的总数。
- `sum(if(qpd.result='right', 1, 0)) as right_question_cnt`：计算每个设备在8月份正确回答的问题数。

2. 数据来源：

```
sql 复制代码
from user_profile as up
```

- 从 `user_profile` 表中选择用户信息。

3. 连接条件：

```
sql 复制代码
left join question_practice_detail as qpd
on qpd.device_id = up.device_id and month(qpd.date) = 8
```

- 将 `question_practice_detail` 表与 `user_profile` 表通过 `device_id` 连接，且筛选出8月份的数据。使用 **LEFT JOIN** 确保即使在 `question_practice_detail` 表中没有对应的记录，也会保留 `user_profile` 表中的记录。

4. 筛选条件：

```
sql 复制代码
where up.university = '复旦大学'
```

- 只选择来自“复旦大学”的用户数据。

5. 分组条件：

```
sql 复制代码
group by up.device_id
```

- 按 `device_id` 对结果进行分组，以便对每个设备分别计算问题总数和正确问题数。

总结

这条查询的核心目的是从“复旦大学”的用户中，统计每个设备在8月份的问题练习情况。通过 `LEFT JOIN` 将 `user_profile` 表与 `question_practice_detail` 表连接，获取每个设备的问题总数和正确回答的问题数。

图 3.14: 核心线索梳理

我们可以从复杂代码的核心线索来更清晰地理解代码逻辑对于逻辑复杂的代码，我们可以让 GPT 在进行代码解释后，进一步梳理出代码流程图，尤其是对于一些算法问题。将提示语句“这个代码的流程比较复杂，请帮我梳理出一条核心的线索，帮助我更好的读懂它”输入给 GPT，这里的关键词是流程和核心线索。

```
#假设是5*5的列表
LENTH = 5
WIDTH = 5
# 先初始化全是0的矩阵
spin = [[0] * LENTH for i in range(WIDTH)]
x = 0
y = 0
count = 1
# 方向 0 向下 1 右 2 上 3 左
direction = 0
#
#while x<LENTH and y<WIDTH:
while True:
    spin[x][y] = count
    count += 1
    if spin[x][y] == LENTH*WIDTH:
        break
    if direction == 0:
        if x == LENTH-1 or spin[x+1][y] != 0:
            direction += 1
            y += 1
            continue #不再判断接下来的elif 直接回到while进行判断
    else:
        x += 1

    elif direction == 1:
        if y == WIDTH-1 or spin[x][y+1] != 0:
            direction += 1
            x -= 1
            continue
    else:
        y += 1
```

```

elif direction == 2:#向上
    if x==0 or spin[x-1][y] != 0:
        direction += 1
        y -= 1
        continue
    else:
        x -= 1

elif direction == 3:
    if spin[x][y-1] != 0:
        direction = 0
        x += 1
        continue
    else:
        y -= 1

for i in spin:
    for j in i:
        print("%02d" % j, ' ' * 2, end="")
    print("")

```

这个代码的流程比较复杂，请帮我梳理出一条核心的线索，帮助我更好的读懂它

取消

发送

- 🌀 这个代码的核心目标是填充一个 5×5 的矩阵，按照螺旋顺序将从1到25的数字填入矩阵中。要更好地理解这个代码，可以从以下几个关键的线索：↓ 理它的流程：

🗣️ 给“ChatGPT”发送消息

ChatGPT 也可能会犯错。请核查重要信息。

图 3.15: 复杂代码

- 🌀 这个代码的核心目标是填充一个 5×5 的矩阵，按照螺旋顺序将从1到25的数字填入矩阵中。要更好地理解这个代码，可以从以下几个关键的线索来梳理它的流程：

核心线索

1. 矩阵的初始化:

```
python 复制代码
spin = [[0] * LENTH for i in range(WIDTH)]
```

这行代码创建了一个 5×5 的矩阵，所有元素初始化为0。

2. 方向的控制:

```
python 复制代码
direction = 0
```

变量 `direction` 用来控制填充数字的方向。`direction` 有四个可能的值:

- `0` 代表 向下 (`x` 增加, `y` 不变)
- `1` 代表 向右 (`x` 不变, `y` 增加)
- `2` 代表 向上 (`x` 减少, `y` 不变)
- `3` 代表 向左 (`x` 不变, `y` 减少)

3. 数字填充的循环:

```
python 复制代码
while True:
    spin[x][y] = count
    count += 1
```

这个循环不断将 `count` 赋值给当前坐标 `(x, y)`，然后递增 `count`。

4. 方向的判断和改变:

```
python 复制代码
if direction == 0:
    if x == LENTH-1 or spin[x+1][y] != 0:
        direction += 1
        y += 1
        continue
    else:
        x += 1
```

每个方向的代码块中包含一个判断条件来检查是否需要改变方向。条件包括:

- 到达边界 (例如, `x == LENTH-1` 代表向下时到达底边)
- 当前单元格已经被填充 (例如, `spin[x+1][y] != 0` 代表下一个单元格已填充)

5. 改变方向:

当满足改变方向的条件时:

- 增加 `direction` 以改变方向 (如 `direction += 1` 从 `0` 变为 `1`)
- 根据新的方向更新 `(x, y)` 以适应新的填充路径

6. 方向变更后更新坐标:

```
python 复制代码
if direction == 0:
    x += 1
elif direction == 1:
    y += 1
elif direction == 2:
    x -= 1
elif direction == 3:
    y -= 1
```

图 3.16: 代码解释

这是一个生成五行五列的螺旋矩阵的 Python 代码，GPT 先以“小标题 + 代码片段 + 代码解释”的方式对代码进行解释，然后梳理出代码的结构图，将代码的流程进行清晰的展现。

通过以上例子可以看出，掌握一些提示语句技巧有助于我们更好地利用大模型来辅助我们解释及改进代码，提高实际编程工作的效率与准确性。在提示语句中，我们可以要求大模型输出内容的具体方式，也可以加上一些提示词来使大模型的回答内容更加准确与完善。在大模型给出解释以及代码后，可以进一步追问，优化代码或对不准确的地方进行提问。为防止代码出现错误和安全隐患，可以要求大模型生成单元测试。除了让大模型解释自己提供的代码，比如转换编程语言，修改部分代码等实际应用场景，还可以提供问题与背景信息，让大模型编写程序并进行解释。

通过本节的学习，读者应能够初步掌握如何有效地利用大模型进行代码解释和优化，也为未来在这一领域的探索提供了思路 and 方向。

3.3 大模型代码纠错

除了解释代码之外，大模型还可以用来进行代码纠错，下面以 GPT 为例，对编程中较为常见的代码错误进行纠正。

```
print(1)
file=open('hahaha98','r+') #先去读一个文件，如果能打开的话就可以写入
print(2)
```

在上述代码的第二行中，我们准备采用“r+”的方式读取并写入文档，但是代码运行后会出现如下错误：

```
Traceback (most recent call last):
  File "c:\Users\DELL\Desktop\代码纠错.py", line 2, in <module>
    file=open('hahaha98','r+') #先去读一个文件，如果能打开的话就可以写入
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'hahaha98'
```

图 3.17: 代码错误 1

下面我们将代码原文以及出现的问题发送给 GPT，并让它用中文对代码存在的问题进行解释，最终输出的结果如下图所示：

```
1 print(1)
file=open('hahaha98','r+') #先去读一个文件，如果能打开的话就可以写入
print(2)

Traceback (most recent call last):
File "c:\Users\DELL\Desktop\代码纠错.py", line 2, in <module>
file=open('hahaha98','r+') #先去读一个文件，如果能打开的话就可以写入
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'hahaha98'
```

用中文回复


 这个错误是因为 Python 无法找到名为 'hahaha98' 的文件。在使用 'r+' 模式打开文件时，Python 要求文件已经存在。如果你希望打开一个文件用于读取和写入，并且如果文件不存在则创建它，可以使用 'w+' 模式。示例如下：



图 3.18: 大模型分析结果 1

通过上图可以看出,大模型顺利识别出当前代码报错的原因是当前并不存在名为“hahaha98”的文件,并通过“try-except”代码结构帮助我们在目标文件不存在时通过“w+”方式创建并读取文档。

大模型不仅可以对报错的代码进行纠错。如果我们尝试运行的代码并未报错,但最终的运行结果并非我们预期的结果也可以通过大模型来查找解决办法。以如下代码为例:

```
current_users = ["A","b","C","d"]
current_users_copy = current_users[:] #将current_users 赋值给 current_users_copy
for value in current_users_copy:
    value=value.lower()
print(current_users_copy)
```

上述代码的目的是在 `current_users` 的基础上生成一个新的 `current_users_copy` 列表,并且新的 `current_users_copy` 列表中的字母完全编程小写。运行上述代码,最终结果如下图所示:

```
PS C:\Users\DELL> & D:/Python3.12/python.exe c:/Users/DELL/Desktop/代码纠错.py
['A', 'b', 'C', 'd']
```

图 3.19: 代码错误 2

可以看出最终输出的结果并未报错,但是也并未能将新生成的 `current_users_copy` 列表中的各个字母元素变为小写,下面我们将源代码以及存在的问题发送给 GPT,最终输出的结果如下图所示:

```

1 current_users = ["A","b","C","d"]
current_users_copy = current_users[:] #将current_users 赋值给current_users_copy
for value in current_users_copy:
value=value.lower()
print(current_users_copy) 为什么current_users_copy中还是有大写字母?

```

问题出在你对列表中的元素进行小写化的方式上。在你的代码中，虽然使用了切片 `current_users_copy = current_users[:]` 来复制列表，但是在遍历并尝试修改 `current_users_copy` 中的元素时，你使用了一个错误的方式。

在 Python 中，当你遍历一个列表并对其元素进行操作时，直接赋值给 `value` 并修改 `value` 并不会影响原始列表中的元素。要修改列表中的元素，你需要通过索引来进行操作，或者使用正确的赋值方式。

这里是一个修正后的例子：

```

python
current_users = ["A", "b", "C", "d"]
current_users_copy = current_users[:] # 将current_users复制给current_users_copy
for i in range(len(current_users_copy)):
    current_users_copy[i] = current_users_copy[i].lower()

print(current_users_copy) # ['a', 'b', 'c', 'd']

```

解释修正后的代码：

1. `current_users_copy = current_users[:]` 使用切片来创建 `current_users` 的副本 `current_users_copy` 。
2. `for i in range(len(current_users_copy)):`
- 使用 `range(len(current_users_copy))` 来获取索引，因为我们需要修改原始列表中的元素。
3. `current_users_copy[i] = current_users_copy[i].lower()`
- 通过索引 `i` 来访问列表中的每个元素，并使用 `.lower()` 方法将其转换为小写字母形式。
4. 最后打印 `current_users_copy` ，你会看到所有元素都已经转换为小写字母形式。

这样，`current_users_copy` 中的元素都会被正确地转换为小写字母，而不会影响到 `current_users` 中的原始元素。

图 3.20：大模型分析结果 2

通过上图可以看出，大模型准确判断出新生成的 `current_users_copy` 中的字母元素之所以未能变成小写，是因为遍历一个列表并对其元素进行操作时，直接赋值给 `value` 并修改 `value` 并不会影响原始列表中的元素。要修改列表中的元素，就需要通过索引来进行操作，或者使用正确的赋值方式。并且大模型还给出了相应的正确可行的代码，并对相应代码进行了细致地解释。除了对简单的代码进行修改之外，大模型还能在神经网络训练等方面根据我们的需求进行合理的代码优化。以如下代码为例：

```

from sklearn.neural_network import MLPClassifier
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

digits =load_digits()#载入数据
x_data= digits.data #数据
ydata=digits.target #标签

#分割数据 1/4为测试数据， 3/4为训练数据
x_train,x_test,y_train,y_test=train_test_split(x_data,y_data)
#64-100-50-10

```

```
#64-100-50-10
mlp=MLPClassifier(hidden_layer_sizes=(100,50),max_iter=500)
mlp.fit(x_train,y_train)
mlp.fit(x_train,y_train)

predictions = mlp.predict(x_test)
print(classification_report(y_test,predictions))
```

上述代码为一个简单的神经网络结构，主要包括载入数据、切分数据、模型构建（包括两个隐藏层）、训练模型、预测结果等部分。如果我们想将上述神经网络结构进行改变，比如：需要搭建有3个隐藏层的网络结构，第1层20个神经元第2层15个神经元，第3层10个神经元，输出有3个分类。那我们就可以直接将源代码和要求发送给GPT，其最终的回答如下图所示：

要修改你的神经网络结构为3个隐藏层，分别有20个、15个和10个神经元，并且输出有3个分类，你可以按照以下方式修改代码：

```
from sklearn.neural_network import MLPClassifier
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

digits = load_digits() # 载入数据
x_data = digits.data # 数据
y_data = digits.target # 标签

# 分割数据1/4为测试数据，3/4为训练数据
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, random_state=42)

# 构建MLPClassifier模型，指定3个隐藏层结构
mlp = MLPClassifier(hidden_layer_sizes=(20, 15, 10), max_iter=500, random_state=42)
mlp.fit(x_train, y_train)

predictions = mlp.predict(x_test)
print(classification_report(y_test, predictions))
```

修改说明：

1. 隐藏层结构修改：
 - `hidden_layer_sizes=(20, 15, 10)`：这里指定了三个隐藏层，分别有20个、15个和10个神经元。
2. 训练数据和测试数据分割：
 - `train_test_split(x_data, y_data, random_state=42)`：使用 `random_state=42` 来确保每次运行时分割的结果一致性。
3. MLPClassifier初始化：
 - `MLPClassifier(hidden_layer_sizes=(20, 15, 10), max_iter=500, random_state=42)`：创建一个MLPClassifier对象，指定了三个隐藏层的结构以及最大迭代次数为500次。

这样修改后，你的神经网络将按照要求的结构进行训练和预测。记得根据具体需求调整随机种子 `random_state`，以便复现结果。

图 3.21：大模型分析结果 3

可以看出，GPT 准确地在需要修改的隐藏层部分进行了修改。此外，我们也可以要求 GPT 对上述代码进行转换。例如，可以将上述代码转换成 R 语言，最终输出的结果如下图所示：

```

Sure, here's the equivalent code in R using the neuralnet package for building a neural network:

# Load necessary libraries
library(neuralnet)
library(datasets)

# Load digits dataset from datasets package
data(digits)
x_data <- as.matrix(digits$data) # Features
y_data <- digits$target # Labels

# Split data into training and testing sets
set.seed(42) # For reproducibility
train_indices <- sample(1:nrow(x_data), 0.75 * nrow(x_data))
x_train <- x_data[train_indices, ]
y_train <- y_data[train_indices]
x_test <- x_data[-train_indices, ]
y_test <- y_data[-train_indices]

# Build neural network with 3 hidden layers: 20, 15, and 10 neurons respectively
nn <- neuralnet(
  y_data ~ .,
  data = cbind(x_train, y_train),
  hidden = c(20, 15, 10),
  linear.output = FALSE # Since it's a classification task
)

# Make predictions on test set
nn_predictions <- predict(nn, x_test)
predicted_classes <- apply(nn_predictions, 1, which.max) - 1 # Convert to 0-based index

# Print classification report
print(table(predicted_classes, y_test))

```

图 3.22: 大模型分析结果 4

并且给出了相应的注释：

```

Explanation:

1. Loading Libraries and Dataset:

• library(neuralnet) : Loads the neuralnet package for neural network operations.
• library(datasets) : Loads the datasets package where the digits dataset is available.

2. Data Preparation:

• data(digits) : Loads the digits dataset.
• x_data and y_data : Extract features and labels from the dataset.
• set.seed(42) : Sets the random seed for reproducibility.
• Splits the data into training ( x_train , y_train ) and testing ( x_test , y_test ) sets.

3. Neural Network Construction:

• neuralnet() : Builds the neural network model with 3 hidden layers (20, 15, 10) using the neuralnet function. The formula y_data ~ . specifies that y_data is predicted by all other variables ( . ).

4. Prediction and Evaluation:

• predict(nn, x_test) : Generates predictions on the test set using the trained neural network nn .
• apply(nn_predictions, 1, which.max) - 1 : Converts the output predictions into 0-based index predictions for comparison with y_test .
• print(table(predicted_classes, y_test)) : Prints a confusion matrix for evaluation.

This R code mirrors the functionality of the Python code provided earlier, adapting it to the syntax and conventions of R programming using the neuralnet package for neural network implementation.

```

图 3.23: 大模型分析结果 5

除了对已有的代码进行纠错外，大模型甚至能够根据我们的需求去设计代码。例如：假设我们有 3 个文件 TS1.txt, TS2.txt, TS3.txt, 每个文件里面都保存了按时间顺序产生出来的 60 个

数值，现在我们要求 GPT 帮我用 python 程序读取这 3 个文件并在同一幅图中把 3 个文件中的数据画出来。

最终 GPT 的回答如下图所示：

1 我有3个文件TS1.txt, TS2.txt, TS3.txt, 每个文件里面都保存了按时间顺序产生出来的60个数值, 帮我用python程序读取这3个文件并在同一幅图中把3个文件中的数据画出来。

Certainly! You can achieve this task in Python using libraries like `matplotlib` for plotting and basic file handling for reading data from text files. Here's a step-by-step approach:

1. Reading Data from Files:

- Read each file (`TS1.txt` , `TS2.txt` , `TS3.txt`).
- Store the data from each file into separate lists.

2. Plotting Data:

- Use `matplotlib` to create a single plot where data from all three files are plotted together.

Here's the Python code to accomplish this:

```
import matplotlib.pyplot as plt

# Function to read data from a file
def read_data(file_name):
    with open(file_name, 'r') as file:
        data = [float(line.strip()) for line in file.readlines()]
    return data

# File names
file_names = ['TS1.txt', 'TS2.txt', 'TS3.txt']

# Read data from each file
data = [read_data(file) for file in file_names]

# Plotting
plt.figure(figsize=(10, 6)) # Adjust size if needed

# Plot each dataset
for i in range(len(data)):
    plt.plot(range(1, len(data[i]) + 1), data[i], label=f'TS{i+1}')

plt.xlabel('Time')
plt.ylabel('Value')
plt.title('Time Series Data from TS1.txt, TS2.txt, TS3.txt')
plt.legend()
plt.grid(True)
plt.tight_layout()

# Display the plot
plt.show()
```

Explanation:

- **Function `read_data`**: Reads data from a specified file (`file_name`) and returns it as a list of floats after stripping any newline characters.
- **File Handling**: Uses a list comprehension to read each file (`TS1.txt` , `TS2.txt` , `TS3.txt`) into separate lists (`data`).
- **Plotting**:
 - Initializes a plot with `plt.figure(figsize=(10, 6))` to adjust the figure size as needed.
 - Uses a loop to plot each dataset (`TS1` , `TS2` , `TS3`) using `plt.plot` .
 - Sets labels, title, legend, and grid for the plot.
 - `plt.tight_layout()` ensures the plot elements are well-spaced.
- **Displaying the Plot**: Finally, `plt.show()` displays the plot on screen.

This script assumes that each file (`TS1.txt` , `TS2.txt` , `TS3.txt`) contains exactly 60 values, one per line. Adjustments can be made to handle different file structures or sizes accordingly.

图 3.24: 大模型分析结果 6

通过上述实例可以看出，GPT 不仅能找出 bug 出现的原因，还能提供修改后的代码，甚至可以直接复制可用。但是，值得注意的是，虽然 GPT 在这些任务上的表现令人印象深刻，但它并不完美。有些复杂的编程错误，可能需要更深的编程知识或对特定代码库的理解，这可能超出了 GPT 的能力范围。

与此同时，我们还需谨记，像 GPT 这样的 AI 模型，虽然在许多任务上的表现都出奇的好，但是对其结果的理解和信任必须建立在广泛的经验和理解上，并且需要明白其潜在的偏差和限制。这可以避免在我们过分依赖这些系统进行决策时，可能带来的一些风险或不必要的复杂性。

AI 正在变革我们解决问题的方式，特别是在涉及到日常任务中如代码调试这样通常需要深入学习和理解才能解决的问题。GPT 在代码错误修复方面的使用，无疑为我们的软件开发流程开辟了新的可能性，并为未来的研究提供了启示。但需要记住的是，正确使用和理解 AI 技术的重要性，不能忽视。



CHAPTER 4

第 4 章 大模型数据分析

4.1 使用 AI 进行简单的数据分析

4.1.1 简介

随着数据量的迅速增长和技术的进步，人工智能（AI）在数据分析领域的应用日益广泛。本节的核心目的是为读者提供如何利用 AI 这一先进的智能助手处理结构化数据的指导。


尽管当前的大型语言模型（如 ChatGPT、Kimi 等）主要侧重于自然语言理解和生成，而非直接执行数值计算或图像处理，但通过其强大的语言处理能力，这些模型能够支持用户解释数据、生成详细的分析报告，并提供对数据趋势和特征的描述。这种能力可以帮助用户从复杂的数据集中提取深刻见解和重要信息，从而在数据分析过程中做出准确的决策和深入的理解。

4.1.2 步骤

以下是使用 AI 进行简单数据分析的基本步骤（以 Kimi 为例）：

所用数据集地址为：https://github.com/aoyinke/ML_learner 在 Kimi 对话框的右侧可以点击上传文件，接受 pdf、doc、xlsx、ppt、txt、图片等。

请你对这个文件中的数据进行数据总览，接着进行数据准备与预处理(包括数据导入、清洗、转换等)，然后进行基本的数据分析（包括描述性统计、分析和数据可视化等，这一部分是主要部分需要详细回答）。最后请附带Python代码。

 **diabetes**
CSV, 23.31 KB

数据总览

首先，我们需要对提供的数据集进行初步的了解。数据集名为diabetes.csv，根据内容，它是一个关于糖尿病患者的医疗记录数据集。每一行代表一个患者的数据，包括以下特征：

- Pregnancies: 怀孕次数
- Glucose: 血糖浓度
- BloodPressure: 血压
- SkinThickness: 皮肤厚度
- Insulin: 胰岛素水平
- BMI: 体重指数
- DiabetesPedigreeFunction: 糖尿病家族史函数值
- Age: 年龄
- Outcome: 结果（1表示患有糖尿病，0表示没有糖尿病）

图 4.1: 提出问题

数据准备与预处理

在进行任何数据分析之前，必须确保数据的完整性和可用性，在向 Kimi 提问时也涉及了这部分。具体包括：

- 数据导入：使用 Python 中的 Pandas 库读取 CSV 文件或处理文本数据，以便后续分析。

数据准备与预处理

数据导入

使用Python的pandas库来导入数据。

```
import pandas as pd

# 读取CSV文件
df = pd.read_csv('diabetes.csv')
```

图 4.2: 数据导入

- 数据清洗：检测并处理数据中的缺失值、异常值和重复项，确保数据质量和准确性。

数据清洗

检查缺失值、异常值和重复数据。

```

# 检查数据类型
print(df.dtypes)

# 检查缺失值
print(df.isnull().sum())

# 检查重复数据
print(df.duplicated().sum())

# 异常值检查 (例如BMI不可能为0)
print(df[df['BMI'] == 0])

```

图 4.3: 数据清洗

- 数据转换：根据分析需求选择并转换数据类型，如数值型、文本型或类别型数据。

数据转换

将Outcome列转换为数值型，1表示阳性，0表示阴性。

```
df['Outcome'] = df['Outcome'].astype(int)
```

图 4.4: 数据转换

利用 AI 模型进行数据分析

使用 AI 模型（如 Kimi）对预处理后的数据进行分析：

- 描述性统计分析：利用 AI 模型执行描述性统计分析，包括平均值、标准差、频率分布等，帮助理解数据的基本特征和分布情况。

describe() 函数是 pandas 库中的一个统计方法，用于生成数据集的基本描述性统计信息。默认输出以下统计指标：计数 (count)、均值 (mean)、标准差 (std)、最小值 (min)、25th 百分位数 (25%)、中位数 (50%)、75th 百分位数 (75%)、最大值 (max)。默认情况下，只会对数值型数据列进行计算，对于非数值型数据列（如字符串或分类变量），将自动忽略。以下是 Kimi 回答给出的代码以及代码运行的结果，可以加入代码导出为 csv 文件形式。

基本的数据分析

描述性统计

使用pandas的描述性统计方法来获取数据的概览。

```
# 描述性统计
desc_stats = df.describe()
print(desc_stats)
```

图 4.5: 描述性统计代码

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin \
count	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479
std	3.369578	31.972618	19.355807	15.952218	115.244002
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000
75%	6.000000	140.250000	80.000000	32.000000	127.250000
max	17.000000	199.000000	122.000000	99.000000	846.000000

	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000
mean	31.992578	0.471876	33.240885	0.348958
std	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.078000	21.000000	0.000000
25%	27.300000	0.243750	24.000000	0.000000
50%	32.000000	0.372500	29.000000	0.000000
75%	36.600000	0.626250	41.000000	1.000000
max	67.100000	2.420000	81.000000	1.000000

图 4.6: df.describe() 函数运行结果

其他信息，如峰度、偏度、频率分布表、相关系数矩阵等，可继续提问获取答案。

比如下图：

- 数据可视化：通过 AI 模型生成数据可视化图像，如折线图、柱状图、散点图、热力图等，以展示数据变化趋势或关联性。下页可见 Kimi 给出的代码示例以及代码运行的结果。

生成数据可视化图像后，需要对其进行评估以确保其准确地反映了数据的变化趋势或关联性。评估方法包括但不限于：

- 视觉评估：通过直观观察和比较，评估生成的图像是否符合预期的数据模式和特征。
- 定量评估：使用统计指标和度量来量化可视化图像的质量和有效性，如相关性系数、信息熵、数据重建误差等。

```

# 峰度和偏度分析
bmi_kurtosis = kurtosis(df['BMI'])
glucose_skewness = skew(df['Glucose'])

print(f"BMI Kurtosis: {bmi_kurtosis}")
print(f"Glucose Skewness: {glucose_skewness}")

# 频率分布表
outcome_freq_dist = df['Outcome'].value_counts()
print(outcome_freq_dist)

# 相关系数矩阵
corr_matrix = df.corr()

# 打印相关系数矩阵
print(corr_matrix)

```

图 4.7: Kimi 提供的相关代码

频率分布表的可视化

我们可以使用matplotlib和seaborn来绘制直方图和累积分布函数 (CDF) 图。

```

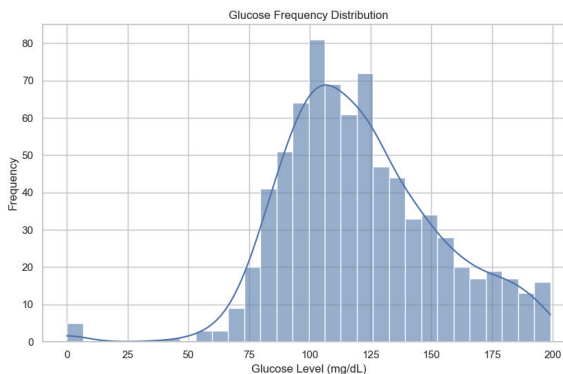
import matplotlib.pyplot as plt
import seaborn as sns

# 绘制Glucose的直方图
plt.figure(figsize=(10, 6))
sns.histplot(df['Glucose'], bins=30, kde=True)
plt.title('Glucose Frequency Distribution')
plt.xlabel('Glucose Level (mg/dL)')
plt.ylabel('Frequency')
plt.show()

# 绘制Glucose的累积分布函数 (CDF) 图
plt.figure(figsize=(10, 6))
sns.ecdfplot(df['Glucose'])
plt.title('Cumulative Distribution Function of Glucose')
plt.xlabel('Glucose Level (mg/dL)')
plt.ylabel('Cumulative Probability')
plt.show()

```

(a) 代码



(b) 图片

图 4.8: Glucose 的可视化

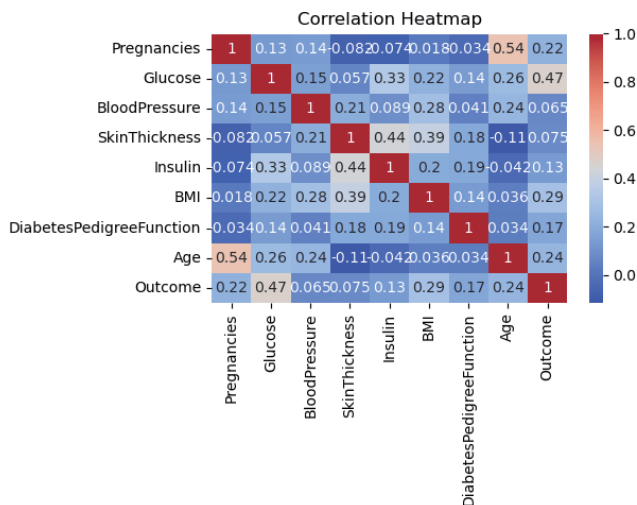


图 4.10: 热力图

```

# 首先, 我们需要确保Outcome是数值型的, 其中1表示患有糖尿病, 0表示没有糖尿病
df['Outcome'] = df['Outcome'].astype(int)

# 计算Outcome与其他因素之间的相关性
correlations = df.corr()['Outcome']

# 接下来, 我们提取Outcome的相关性, 然后对其进行排序
outcome_corr = correlations.drop('Outcome').sort_values(ascending=False)

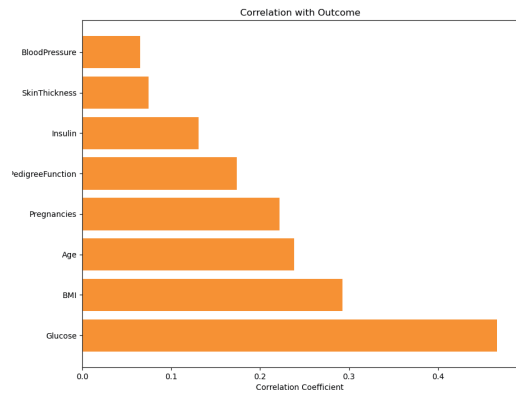
# 绘制水平条形图
fig = plt.figure(figsize=(10, 8)) # 您可能需要调整尺寸以适应图表
plt.barh(range(len(outcome_corr)), outcome_corr, color="#FF9912")
plt.yticks(range(len(outcome_corr)), outcome_corr.index)
plt.title('Correlation with Outcome')
plt.xlabel('Correlation Coefficient')
plt.ylabel('Features')
plt.show()

```

这段代码首先计算了Outcome与其他所有变量之间的相关系数, 然后删除了与自身 (Outcome) 的相关性 (因为总是为1), 并按相关系数的绝对值降序排序。之后, 使用matplotlib绘制了一个水平条形图, 展示了每个因素与Outcome的相关性大小。

请注意, 相关系数的值范围在-1到1之间, 其中1表示完全的正相关, -1表示完全的负相关, 0表示没有线性相关。在条形图中, 颜色较深的条形表示较高的正相关或负相关性。

(a) 代码



(b) 水平条形图

图 4.9: 相关性分析

结果解释与应用

最后, 对 AI 生成的分析结果进行解释和应用:

- **结果验证**: 验证通过 AI 生成的分析结果是否准确地反映了原始数据的特征和趋势。这一步是确保数据分析结果可信和有效的重要环节。

- **洞察发现与应用**: 根据分析结果, 从中提取有用的洞察和见解, 帮助做出决策或进行进一步的数据探索。

- **关键发现的详细解释**: 解释 AI 模型发现的数据模式、趋势或者关系, 例如, 分析结果显示的重要数据点或者变量之间的相关性。

- **决策支持**: 利用提取的洞察, 帮助决策者理解数据背后的含义, 制定更有效的战略或业务策略。

- **辅助分析的应用**: 将先前的描述性统计结果或其他数据分析输出作为输入, 通过 AI 模型 (如 Kimi) 进行更深入的数据挖掘和解释, 揭示隐藏在数据背后的复杂模式和关系。

对于上面案例, 可以将代码运行得到的数据文本、csv 文件、图片等再次输入给 Kimi, Kimi 可以辅助进行结果分析。

比如将上面描述性统计输出的 csv 文件以及热力图再次输入给 Kimi, Kimi 给出了以下分析与解释。

结果解释

- 数据集中的样本数量是768，每个特征都有足够的样本量来代表总体。
- 血糖和胰岛素的标准差较大，表明这两个指标在不同个体间的差异性很大，可能与糖尿病的发病机制密切相关。
- 糖尿病家族史函数值（DiabetesPedigreeFunction）的均值接近0.5，表明在这些样本中，家族史对糖尿病的影响是显著的，但个体间差异较大。
- 年龄的均值和标准差表明，样本涵盖了从年轻到中年的成年人，这有助于研究年龄对糖尿病风险的影响。
- BMI的均值较高，表明超重或肥胖在样本中普遍存在，这与糖尿病的高发有关。

结论

描述性统计结果提供了对数据集特征的初步了解，揭示了可能影响糖尿病风险的因素，如血糖水平、BMI和家族史。这些信息对于进一步的数据分析、建模和糖尿病风险评估至关重要。

图 4.11: Kimi 对描述性统计结果的分析



根据提供的图片结果“图13.png”，这似乎是一个相关系数热力图，显示了糖尿病数据集中各个变量之间的相关性。热力图通常用来可视化两个变量之间的相关系数矩阵，颜色深浅表示相关性的强弱，通常使用蓝色表示负相关，红色表示正相关，颜色的强度表示相关系数的大小。

从图片中，我们可以分析以下几点：

1. **对角线上的值**：对角线上的值始终为1.0，因为这是变量与自身之间的相关性。
2. **Pregnancies（怀孕次数）**：与其他变量的相关性较弱，除了与Glucose（血糖）有轻微的正相关外，与其他变量的相关性接近0或非常小。
3. **Glucose（血糖）**：与BloodPressure（血压）和BMI（体重指数）有轻微的正相关，与Insulin（胰岛素）有中等程度的正相关，这可能表明血糖水平较高的个体可能血压和体重指数也较高，并且胰岛素水平也较高。
4. **BloodPressure（血压）**：与SkinThickness（皮肤厚度）和Insulin（胰岛素）有轻微的正相关，但与BMI（体重指数）有很强的正相关，这符合常识，因为超重或肥胖往往与高血压有关。
5. **SkinThickness（皮肤厚度）**：与Insulin（胰岛素）有轻微的正相关。
6. **Insulin（胰岛素）**：与BMI（体重指数）有很强的正相关，这表明胰岛素水平可能与体重指数有关。
7. **BMI（体重指数）**：与DiabetesPedigreeFunction（糖尿病家族史函数值）和Age（年龄）有轻微的正相关。
8. **DiabetesPedigreeFunction（糖尿病家族史函数值）**：与Age（年龄）有轻微的正相关。
9. **Age（年龄）**：与Outcome（结果）有轻微的负相关，这可能表明年龄较大的人在数据集中可能更少被标记为糖尿病结果阳性。
10. **Outcome（结果）**：与Glucose（血糖）、BMI（体重指数）、Insulin（胰岛素）有轻微的正相关，这符合预期，因为这些因素都与糖尿病风险增加有关。

结论

- 血糖（Glucose）、体重指数（BMI）和胰岛素（Insulin）是与糖尿病结果（Outcome）正相关的因素，其中胰岛素和BMI的相关性较强。
- 血压（BloodPressure）与体重指数（BMI）有很强的相关性，这与医学知识相符。
- 怀孕次数（Pregnancies）似乎与其他变量的相关性较弱，这可能表明它不是糖尿病风险的一个主要因素。
- 年龄（Age）与糖尿病结果（Outcome）有轻微的负相关，这可能是一个需要进一步研究的发现。

图 4.12: Kimi 对热力图的分析

• 报告生成与应用：利用 AI 辅助生成的报告或摘要，将数据分析结果以清晰和易懂的方式呈现给读者。比如上面的案例，Kimi 可以结合所问的问题以及提供的文件、图片等，输出一份简单报告。



糖尿病数据集分析报告

概述

本报告旨在分析糖尿病数据集，以识别影响糖尿病风险的关键因素，并评估不同变量之间的相关性。数据集包含了多个特征，如怀孕次数、血糖、血压、皮肤厚度、胰岛素水平、体重指数（BMI）、糖尿病家族史函数值和年龄，以及一个二元结果标签，指示个体是否患有糖尿病。

数据集概览

数据集包含768个样本，每个样本具有以下特征：

- **Pregnancies**: 平均怀孕次数为3.84次。
- **Glucose**: 平均血糖水平为120.89 mg/dL。
- **BloodPressure**: 平均血压为69.11 mmHg。
- **SkinThickness**: 平均皮肤厚度为20.54 mm。
- **Insulin**: 平均胰岛素水平为79.79 μ IU/mL。
- **BMI**: 平均体重指数为31.99，表明样本中有超重或肥胖的情况。
- **DiabetesPedigreeFunction**: 平均糖尿病家族史函数值为0.47。
- **Age**: 平均年龄为33.24岁。

描述性统计分析

通过计算描述性统计量，我们得到了以下结果：

- 血糖（Glucose）和BMI是与糖尿病结果（Outcome）相关性最高的两个变量。
- 年龄（Age）与糖尿病结果呈现轻微的负相关，这可能表明在数据集中，年龄较大的人可能更少被标记为糖尿病结果阳性。

相关性分析

相关性分析揭示了以下模式：

- 血糖（Glucose）与胰岛素（Insulin）呈现正相关（0.331），表明血糖水平较高的个体可能胰岛素水平也较高。
- BMI与血压（BloodPressure）有较强的正相关（0.282），这符合医学上对超重或肥胖与高血压关联的认识。
- 年龄（Age）与怀孕次数（Pregnancies）有较强的正相关（0.544），表明年龄较大的个体可能有更多的怀孕次数。

数据可视化

通过相关系数热力图，我们可视化了变量之间的相关性。热力图显示，血糖、BMI和胰岛素水平与糖尿病结果有显著的正相关性。此外，年龄与糖尿病结果的相关性较低。

结论与建议

- 血糖水平和BMI是糖尿病风险的显著预测因子。
- 血压和胰岛素水平虽然与糖尿病结果相关，但相关性较弱。
- 年龄与糖尿病结果的负相关性可能表明在样本中，年轻个体更可能被诊断为糖尿病。
- 怀孕次数与年龄的正相关性可能表明生育史对糖尿病风险有潜在影响。

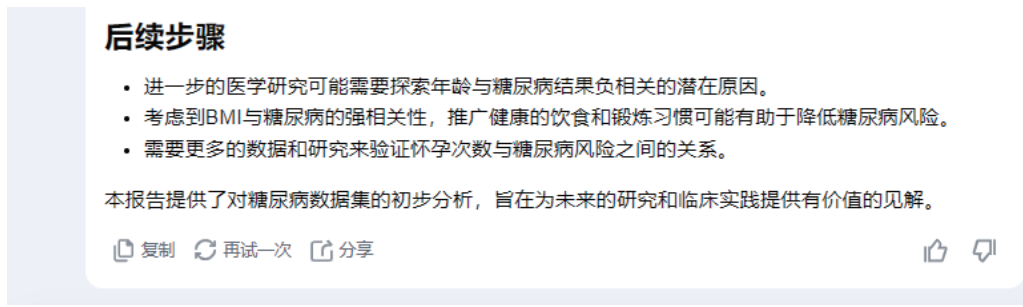


图 4.13: Kimi 生成的报告

通过以上步骤，读者可以学习如何有效地利用 AI 技术进行数据分析，从而在处理大量数据时提高分析的效率和精度。

4.2 使用 AI 完成数学建模项目

上一章节，本书介绍了如何利用 AI 工具完成简单的数据分析，在实际项目中，我们不仅需要对数据进行深层次的挖掘，还需要基于项目目的（如分类、回归问题）对原始数据进行特征工程。所谓特征工程，包括但不限于对数据所给的特征使用相关性检验、共线性检验等一系列方法对变量进行筛选、对原始特性进行组合得到组合特征、对类别型变量进行独热编码等。接下来，本章将以乳腺癌数据集为例，尝试利用 ChatGPT 等 AI 工具完成数据预处理和建模过程。

数据集地址：<https://www.kaggle.com/datasets/reihanenamdari/breast-cancer>

特征及其解释如下：

特征	描述
Age	年龄
Race	种族：白人，黑人，其他（美洲印第安人 / 阿拉斯加土著、亚洲 / 太平洋岛民）
Marital Status	婚姻状况：已婚，单身，离婚，分居，丧偶
T Stage	T 分期：肿瘤直径为依据
N Stage	N 分期：局部淋巴结转移情况为依据
6th Stage	AJCC 第六版分期：原位癌、肿瘤大小、局部淋巴结转移和器官转移等为依据
differentiate	分化程度：未分化，低分化，中等分化，高分化
Grade	等级：高等级意味着癌症增长更快，更容易扩散
A Stage	A 分期：局部（未转移），远处（转移）
Tumor Size	肿瘤大小
Estrogen Status	雌激素状态：ER 阳性乳腺癌
Progesterone Status	孕激素状态：PR 阳性乳腺癌
Regional Node Examined	检查的区域节点：记录病理检查的局部淋巴结总数
Regional Node Positive	区域节点阳性数：记录检查局部阳性淋巴结的数量
Survival Months	生存时间（月）
Status	结局状态：存活，死亡

4.2.1 数据预处理

使用 AI 协助数学建模与数据分析时，提问时需要注意的问题

- 问题具体明确：提出的问题应当明确且具体，避免过于笼统。例如，与其问“如何进行数据预处理？”，不如问“如何使用 Python 处理数据中的缺失值？”。

- 提供足够的上下文：在提问时，提供足够的背景信息和上下文，使 ChatGPT 能够更准确地理解问题。例如，说明数据的来源、类型、规模等信息。

- 逐步提问：将复杂问题分解为多个简单的问题，逐步提问。这样不仅有助于 ChatGPT 理解问题，也能更清晰地获得每一步的解决方案。

- 检查生成的代码和方法：对 ChatGPT 生成的代码和方法要仔细检查，确保其符合预期，并且与实际数据和问题相匹配。不要盲目复制和粘贴代码，应当理解每一行代码的功能和用途。

应用 AI 提供的方法和代码时的注意事项

- 验证生成的代码：对 ChatGPT 生成的代码进行验证和测试，确保其能够正确运行并且产生预期的结果。可以通过小规模数据集进行测试，逐步验证代码的功能。

- 代码优化和调整：根据实际需要对生成的代码进行优化和调整。ChatGPT 生成的代码可能不是最优的方案，需要根据具体情况进行调整和优化。

- 处理潜在错误和异常：ChatGPT 生成的代码可能会有错误或无法处理某些异常情况。在应用代码时，要考虑可能出现的错误和异常情况，并进行相应的处理和改进。

- 理解算法和模型的原理：理解 ChatGPT 提供的算法和模型的原理，确保其适用于当前问题。例如，在使用特定的机器学习算法时，了解其适用范围、优缺点以及可能的调优方法。

- 数据隐私和安全：确保在数据处理和建模过程中，遵循数据隐私和安全的原则。不应将敏感数据直接输入到 ChatGPT 中，必要时进行数据脱敏处理。

- 保持批判性思维：保持批判性思维，不盲目信任 ChatGPT 提供的答案和方法。可以通过查阅相关文献和资料，进一步验证和确认其可行性和正确性。

- 持续学习和改进：持续学习和改进自己的技能和知识，跟踪领域内的最新研究和技术发展。

ChatGPT 是一个工具，应该在掌握基本原理和方法的基础上，灵活应用和改进。

用 AI 协助数据分析与数学建模的具体步骤

- 问题定义

- 目标：确定要解决的具体问题或回答的具体问题。

- 步骤：

- * 明确目标：例如，预测销售额、分类客户群、优化供应链等
- * 设定范围：定义问题的边界和范围，明确哪些因素和变量是相关的。
- * 理解背景：收集和分析与问题相关的背景信息和先前研究。
- 数据收集
 - 目标：收集与问题相关的、足够量的数据。
 - 步骤：
 - * 识别数据源：确定数据的来源，如数据库、在线资源、实验数据等。
 - * 数据采集：通过 SQL 查询、API 调用、网络爬虫或手动输入等方式收集数据。
 - * 数据存储：将收集到的数据存储到合适的存储系统中，如数据库、数据仓库或数据湖。
- 数据预处理
 - 目标：清理和整理数据，使其适合建模。
 - 步骤：
 - * 数据清洗：处理缺失值（插补、删除）、异常值（检测、处理）和重复数据。
 - * 数据转换：将数据标准化（如缩放到 0-1 区间）、规范化（如转换成正态分布）或编码（如将分类变量转换为数值）。
 - * 数据集成：将来自不同来源的数据集成到一个统一的数据集中。
- 特征选择与提取
 - 目标：选择和提取对模型有用的特征（变量）。
 - 步骤：
 - * 特征选择：选择与目标变量相关的特征，可以使用统计方法（如相关系数）或机器学习方法（如特征重要性评估）。
 - * 特征提取：从原始数据中提取新特征，如从时间戳中提取日期和时间，或通过主成分分析（PCA）减少维度。
 - * 特征工程：创建新的特征或变换现有特征以增强模型性能。
- 模型选择
 - 目标：根据问题类型选择合适的 AI 模型。
 - 步骤：
 - * 分析问题类型：确定问题是回归、分类、聚类还是其他类型。
 - * 选择模型：根据问题类型选择合适的模型，如线性回归、逻辑回归、决策树、K 均值聚类等。

- * 考虑复杂性和可解释性：平衡模型的复杂性和可解释性，选择适合实际应用的模型。

- 模型训练

- 目标：使用训练数据拟合模型。

- 步骤：

- * 分割数据：将数据分为训练集和测试集，通常按 7:3 或 8:2 比例分割。

- * 训练模型：使用训练集数据训练模型，通过优化算法（如梯度下降）调整模型参数。

- * 交叉验证：使用交叉验证（如 K 折交叉验证）评估模型的稳定性和泛化能力。

- 模型评估

- 目标：评估模型在测试数据上的性能。

- 步骤：

- * 选择评估指标：根据问题类型选择合适的评估指标，如均方误差（MSE）、准确率、精确率、召回率、F1 分数等。

- * 评估模型：使用测试集数据评估模型性能，通过计算评估指标了解模型的表现。

- * 可视化评估结果：使用混淆矩阵、ROC 曲线等可视化工具展示评估结果。

- 模型优化

- 目标：优化模型以提高其性能。

- 步骤：

- * 调整参数：通过网格搜索、随机搜索或贝叶斯优化等方法调整模型参数。

- * 选择特征：重新评估特征的重要性，移除不相关或冗余特征。

- * 正则化：使用 L1 或 L2 正则化防止模型过拟合。

- * 模型集成：使用集成方法（如随机森林、梯度提升、XGBoost）组合多个模型以提高性能。

- 模型部署

- 目标：将优化后的模型应用于实际问题中。

- 步骤：

- * 选择部署平台：选择合适的部署平台，如云服务（AWS、Azure）、本地服务器或边缘设备。

- * 模型导出：将模型导出为可部署的格式，如 PMML、ONNX、TensorFlow Saved-Model 等。

- * 部署模型：在选定平台上部署模型，通过 API 或批处理系统提供预测服务。

- * 监控模型：设置监控系统，实时监控 API 调用和模型性能。

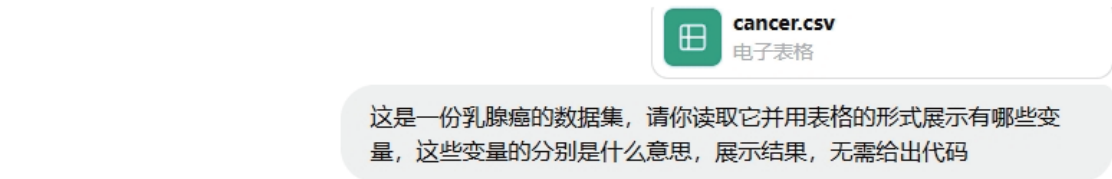
- 模型维护

- 目标：维护和更新模型，确保其在实际应用中的长期有效性。

– 步骤：

- * 定期评估：定期评估模型性能，使用新数据进行再训练。
- * 模型更新：根据评估结果和新数据更新模型，确保其适应新的数据模式和变化。
- * 反馈机制：建立反馈机制，收集用户反馈和模型预测结果，持续改进模型。

首先，使用 ChatGPT 读取乳腺癌 csv 文件并介绍文件中的变量，如图 4.14 所示



数据集中包含以下变量及其意义：

变量名	意义
Age	患者年龄
Race	患者种族
Marital Status	婚姻状况
T Stage	肿瘤大小分期 (T阶段)
N Stage	淋巴结转移分期 (N阶段)
6th Stage	第六版分期
differentiate	分化程度
Grade	分级
A Stage	远处转移分期
Tumor Size	肿瘤大小 (毫米)
Estrogen Status	雌激素受体状态
Progesterone Status	孕激素受体状态
Regional Node Examined	检查的区域淋巴结数量
Reginol Node Positive	阳性区域淋巴结数量
Survival Months	存活月份数
Status	生存状态 (Alive/Dead)

图 4.14: 变量描述

图 4.15 展示了数据预处理步骤中的缺失值检查，通过打印数据集中缺失值的数量来确保数据的完整性。在数据分析和建模过程中，数据的完整性和准确性至关重要。缺失值是指在数据集中某些字段没有记录数据，这可能是由于数据收集过程中的疏漏或数据输入错误造成的。缺失值的存在会对模型的训练和预测产生负面影响，导致模型的准确性降低。因此，在进行数据

分析之前，必须对数据集进行检查，识别并处理缺失值。图中展示了使用 Python 代码检查数据集中每个字段的缺失值数量，通过对这些缺失值进行处理，如删除含有大量缺失值的记录或使用合理的插值方法填补缺失值，可以提高数据的质量和模型的性能。确保数据的完整性是构建可靠预测模型的基础。

首先检查数据中是否存在缺失值和异常值，处理缺失值和异常值是数据清洗的重要步骤。

```
python 复制代码

# 检查缺失值
print(data.isnull().sum())
```

2. 数据编码

将分类变量转换为数值形式，以用于机器学习模型。使用LabelEncoder对所有非数值列进行编码。

```
python 复制代码

from sklearn.preprocessing import LabelEncoder

# 创建一个LabelEncoder对象
labelencoder = LabelEncoder()

# 对所有非数值列进行编码
for column in data.select_dtypes(include=['object']).columns:
    data[column] = labelencoder.fit_transform(data[column])
```

3. 数据标准化

将特征标准化，以消除量纲差异。

```
python 复制代码

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
data_scaled = pd.DataFrame(data_scaled, columns=data.columns)
```

特征选择与提取

通过相关性分析选择重要特征。

请提供模型性能的详细报告。

能否对不同模型进行比较？

图 4.15: 数据预处理 - 检查缺失值

接着可视化变量的相关性，利用 ChatGPT 获取相应代码，运行 Python 代码得到图 4.17 如下：

请用seaborn画出变量间的相关系数热力图，给出可运行的Python代码。



下面是用`seaborn`画出乳腺癌数据集中变量间的相关系数热力图的可运行Python代码：

```
python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# 读取数据集
file_path = '/mnt/data/cancer.csv'
data = pd.read_csv(file_path)

# 计算相关系数矩阵
correlation_matrix = data.corr()

# 设置绘图尺寸
plt.figure(figsize=(12, 10))

# 使用seaborn绘制热力图
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')

# 显示图像
plt.title('Correlation Matrix Heatmap')
plt.show()
```

图 4.16: 热力图代码

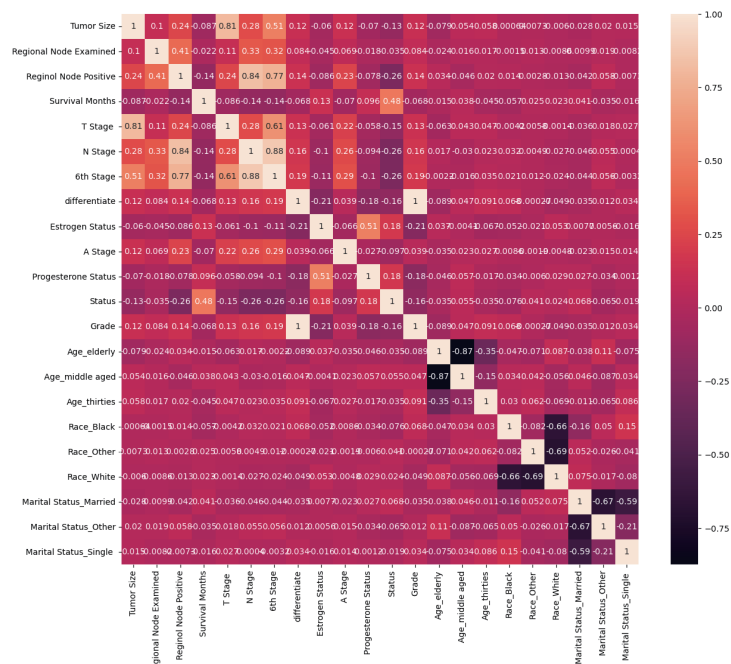


图 4.17: 热力图

通过相关性分析选择重要特征。

```
python 复制代码

import seaborn as sns
import matplotlib.pyplot as plt

# 计算相关系数矩阵
corr_matrix = data.corr()

# 绘制热力图
plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.show()
```

模型选择

我们选择随机森林算法进行模型训练。随机森林是一种集成学习方法，具有良好的分类性能。

模型训练

1. 数据分割

将数据集划分为训练集和测试集，以评估模型的性能。

```
python 复制代码

from sklearn.model_selection import train_test_split

# 假设 'Status' 是目标变量
X = data.drop('Status', axis=1)
y = data['Status']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

请提供模型性能的详细报告。 能否对不同模型进行比较?

图 4.18: 数据编码 - 使用 LabelEncoder 进行编码

图 4.18 展示了将分类变量转换为数值格式的过程，使用了 LabelEncoder 对非数值列进行编码，以便用于机器学习模型。在数据集中，许多变量可能是分类变量（categorical variables），例如种族、婚姻状况等。这些分类变量需要转换为数值格式，以便能够被大多数机器学习算法处理。LabelEncoder 是一种常用的编码方法，可以将分类变量转换为整数值。图中展示了如何使

用 `LabelEncoder` 对数据集中的分类变量进行编码。

首先，创建一个 `LabelEncoder` 对象，然后对每一个非数值列进行编码，将分类变量转换为相应的整数值。这一步骤有助于将数据集中的非数值数据转换为可以被模型处理的数值数据，从而提高模型的训练效果和预测准确性。正确处理分类变量是数据预处理中的关键步骤。

对数据进行缺失值处理、异常值识别处理以及特征工程后，接下来利用 ChatGPT 开始进行数据建模。

4.2.2 数据建模

我们将基于给定的数据集进行一个完整的建模过程。这个数据集包含了癌症患者的各种信息，包括年龄、种族、婚姻状况、癌症分期、肿瘤大小、激素状态、区域淋巴结检查结果和生存状态等。我们将通过以下步骤来完成建模过程。

以 ChatGPT 为例完成数学建模项目

	Age	Race	Marital Status	T Stage
1	68	White	Married	T1
2	50	White	Married	T2
3	58	White	Divorced	T3
4	58	White	Married	T1

对这个数据进行建模，给出具体的建模过程和说明，内容详实，中文作答

具体的建模过程和说明

问题定义

我们希望预测癌症患者的生存状态（存活或死亡）。目标是基于给定的特征（如年龄、种族、肿瘤阶段等）构建一个分类模型，能够准确预测患者的生存状态。

数据收集

数据集已提供，包含了患者的各种信息及其生存状态。我们将此数据集用于模型的训练和评估。

数据预处理

1. 数据清洗

首先检查数据中是否有缺失值。请提供模型性能的详细报告。决定是否对不同模型进行比较？重要步骤。

图 4.19: 癌症数据集示例

2. 训练模型

```
python 复制代码

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# 创建随机森林分类器
model = RandomForestClassifier(n_estimators=100, random_state=42)

# 训练模型
model.fit(X_train, y_train)
```

模型评估

通过测试集评估模型性能，计算准确率、精确率、召回率和F1得分等指标。

```
python 复制代码

# 预测
y_pred = model.predict(X_test)

# 评估模型
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print(f"Classification Report:\n{report}")
```

模型优化

通过网格搜索等方法对模型进行优化，调整模型参数以提高性能。

```
python 复制代码

from sklearn.model_selection import GridSearchCV

提供模型性能的详细报告。 能否对不同模型进行比较？
```

图 4.20: 数据标准化 - 使用 StandardScaler 进行标准化

图 4.20 展示了数据标准化的过程，使用 StandardScaler 对特征进行标准化，以消除量纲差异，提升模型性能。在数据分析和建模过程中，不同特征的取值范围可能会有很大的差异，例如年龄的取值范围可能在几岁到几十岁之间，而肿瘤大小的取值范围可能在几毫米到几厘米之间。这样的量纲差异可能会导致模型在训练过程中倾向于某些取值较大的特征，从而影响模型的训练效果和预测性能。数据标准化是一种常用的方法，通过对每个特征进行标准化处理，使其均值为 0，标准差为 1，从而消除不同特征之间的量纲差异。图中展示了如何使用 StandardScaler 对数据进行标准化，首先创建一个 StandardScaler 对象，然后对数据进行拟合和转换。这一步骤

有助于提高模型的稳定性和准确性，使模型对每个特征的变化都能够有更好的响应。

模型优化

通过网格搜索等方法对模型进行优化，调整模型参数以提高性能。

```
python
from sklearn.model_selection import GridSearchCV

# 定义参数网格
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# 创建GridSearchCV对象
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=3, n_jobs=-1, verbose=1)

# 训练模型
grid_search.fit(X_train, y_train)

# 最佳参数
best_params = grid_search.best_params_

print(f"Best Parameters: {best_params}")
```

模型部署

将训练好的模型部署到生产环境中，通常需要将模型保存为文件，然后加载模型进行预测。

```
python
import joblib
```

↓

请提供模型性能的详细报告。 能否对不同模型进行比较?

保存模型

图 4.21: 特征选择 - 计算相关矩阵并绘制热图

图 4.21 展示了特征选择的过程，通过计算相关矩阵并绘制热图，识别重要特征，降低模型复杂度。在数据分析和建模中，并不是所有的特征都是同等重要的，有些特征对预测目标的影响更大，有些则可能没有太大作用，甚至可能引入噪音，降低模型的性能。特征选择是指从原始特征集中选择对预测目标最有用的特征，以提高模型的准确性和效率。图中展示了如何计算特征之间的相关矩阵，并通过绘制热图来直观地展示特征之间的相关性。通过观察热图，我们可以识别哪些特征与预测目标有较强的相关性，并选择这些特征作为模型的输入。同时，特征

选择还可以帮助我们去除冗余特征，减少模型的复杂度，提高模型的训练速度和预测性能。正确的特征选择是构建高效模型的重要步骤。

```
# 最佳参数
best_params = grid_search.best_params_

print(f"Best Parameters: {best_params}")
```

模型部署

将训练好的模型部署到生产环境中，通常需要将模型保存为文件，然后加载模型进行预测。

```
python 复制代码

import joblib

# 保存模型
joblib.dump(model, 'random_forest_model.pkl')

# 加载模型
loaded_model = joblib.load('random_forest_model.pkl')
```

模型维护

模型部署后，需要定期监控模型的性能，更新模型以应对数据的变化和环境的变化。

1. **性能监控**: 定期评估模型在新数据上的表现。
2. **模型更新**: 根据需要重新训练模型。
3. **故障排除**: 及时发现和解决模型在生产环境中的问题。

通过以上步骤，我们完成了从问题定义到模型维护的完整过程。包括数据的收集、预处理、特征选择与提取、模型选择、训练、评估、优化、部署和维护等每一个环节。 [↩]

🔊 📄 🔄 📌 ✨

请提供模型性能的详细报告。

能否对不同模型进行比较?

图 4.22: 模型训练 - 使用 RandomForestClassifier 进行训练

图 4.22 展示了模型训练的过程，使用随机森林分类器（RandomForestClassifier）对数据进行训练，并评估模型性能。在机器学习中，模型训练是指利用数据集对模型参数进行调整，使模型能够从数据中学习规律，从而对新数据进行准确预测。随机森林是一种集成学习方法，通过构建多个决策树，并对每个决策树的预测结果进行投票表决，从而提高模型的泛化能力和稳定性。图中展示了如何使用 Python 的 sklearn 库中的 RandomForestClassifier 对数据进行训练。

首先，创建一个 `RandomForestClassifier` 对象，设置相关参数，然后对训练数据进行拟合，训练模型。训练完成后，我们可以使用测试数据对模型进行评估，计算模型的准确性、精确率、召回率和 F1 得分等指标。这些评估指标有助于我们了解模型的性能，判断模型是否能够有效地解决实际问题。通过合理的模型训练和评估，我们可以构建一个准确且可靠的预测模型。

ChatGPT 提供的代码及运行结果

检查缺失值、数值类型转化和数据标准化的代码如下。

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler

# 加载数据
file_path = r'C:/Users/13597/Desktop/cancer.csv' # 修改为你的文件路径
data = pd.read_csv(file_path)

# 检查数据
print(data.head())

# 检查缺失值
print(data.isnull().sum())

# 将分类变量转换为数值类型
labelencoder = LabelEncoder()
for column in data.select_dtypes(include=['object']).columns:
    data[column] = labelencoder.fit_transform(data[column])

# 数据标准化
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
data_scaled = pd.DataFrame(data_scaled, columns=data.columns)

# 检查标准化后的数据
print(data_scaled.head())
```

数据前五:

	Age	Race	Marital Status	...	Reginol	Node	Positive	Survival	Months	Status
0	68	White	Married	...			1		60	Alive
1	50	White	Married	...			5		62	Alive
2	58	White	Divorced	...			7		75	Alive
3	58	White	Married	...			1		84	Alive
4	47	White	Married	...			1		50	Alive

[5 rows x 16 columns]

缺失值情况:

Age	0
Race	0
Marital Status	0
T Stage	0
N Stage	0
6th Stage	0

```

differentiate      0
Grade              0
A Stage           0
Tumor Size        0
Estrogen Status   0
Progesterone Status 0
Regional Node Examined 0
Regional Node Positive 0
Survival Months   0
Status            0
dtype: int64

```

标准化后的数据前五:

```

      Age      Race  ...  Survival Months  Status
0  1.565253  0.39716  ...      -0.492961 -0.425148
1 -0.443222  0.39716  ...      -0.405695 -0.425148
2  0.449434  0.39716  ...       0.161530 -0.425148
3  0.449434  0.39716  ...       0.554224 -0.425148
4 -0.777968  0.39716  ...      -0.929288 -0.425148

```

```
[5 rows x 16 columns]
```

图 4.23: 检查缺失值、数据类型转化和数据标准化代码运行结果

模型训练与评估的代码如下。

```

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

x = data.drop('Status', axis=1)
y = data['Status']

# 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 训练模型
model.fit(X_train, y_train)

# 预测
y_pred = model.predict(X_test)

# 评估模型
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print(f"Classification Report: \n{report}")

```

```

Accuracy: 0.9130434782608695
Classification Report:

```

	precision	recall	f1-score	support
0	0.92	0.98	0.95	685
1	0.83	0.53	0.64	120
accuracy			0.91	805
macro avg	0.88	0.75	0.80	805
weighted avg	0.91	0.91	0.90	805

```
Process finished with exit code 0
```

图 4.24: 训练与评估代码运行结果

模型优化和部署代码如下。

```

from sklearn.model_selection import GridSearchCV
import joblib

# 定义参数网络
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# 创建GridSearchCV对象
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=3, n_jobs=-1, verbose=2
                           )

# 训练模型
grid_serach.fit(X_train, y_train)

# 最佳参数
best_params = grid_search.best_params_

print(f"Best Parameters: {best_params}")

# 保存模型
joblib.dump(model, 'random_forest_model.pkl')
# jiazaimoxing
loaded_model = joblib.load('random_forest_model.pkl')

```



```

Fitting 3 folds for each of 108 candidates, totalling 324 fits
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=2, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=2, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=2, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=10, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=10, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=10, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=2, n_estimators=100; total time= 0.3s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=2, n_estimators=100; total time= 0.3s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=2, n_estimators=100; total time= 0.3s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=5, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=5, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=10, n_estimators=100; total time= 0.2s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=10, n_estimators=100; total time= 0.2s
[CV] END max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time= 0.6s
[CV] END max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=50; total time= 0.0s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time= 0.6s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=5, n_estimators=50; total time= 0.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=5, n_estimators=100; total time= 0.2s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=10, n_estimators=200; total time= 0.5s
[CV] END max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=100; total time= 0.2s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=5, n_estimators=100; total time= 0.2s
[CV] END max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=100; total time= 0.2s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time= 0.6s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=10, n_estimators=200; total time= 0.5s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=10, n_estimators=200; total time= 0.5s
[CV] END max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=100; total time= 0.2s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=5, n_estimators=100; total time= 0.2s

```

图 4.25: 模型优化和部署代码运行结果

以上介绍了使用人工智能（AI）协助数学建模的详细步骤。首先，在问题定义阶段，AI 可以帮助明确建模目标。接着，数据收集阶段，AI 可以自动化数据的收集和整理。数据预处理阶段，AI 能高效地进行数据清洗、编码和标准化处理。特征选择与提取阶段，AI 通过相关性分析和特征重要性评估，快速选出关键特征。模型选择阶段，AI 推荐并优化适合的机器学习算法。模型训练阶段，AI 能自动调参并加速训练过程。模型评估阶段，AI 提供准确的评估指标如准确率、精确率、召回率和 F1 得分等。最后，在模型优化和部署阶段，AI 帮助调整参数提高模型性能，并在实际场景中应用和维护模型。通过这些步骤，AI 显著提升了数学建模的效率和精确度。



CHAPTER 5

第 5 章 大模型科学研究

5.1 大模型论文搜索

5.1.1 大模型的原理

大语言模型通常基于深度神经网络,尤其是卷积神经网络(CNNs)和循环神经网络(RNNs)。近年来,基于 Transformer 架构的模型(如 GPT 和 BERT)变得特别流行。

神经网络由多个层组成,每层包含多个神经元。每个神经元接受输入,进行加权求和并通过激活函数生成输出。

输入数据经过各层的逐步处理,最终产生输出结果。

通过反向传播算法调整网络权重,以最小化损失函数。训练数据和优化算法(如梯度下降)用于此过程。

大语言模型使用大量文本数据进行训练。数据集通常包括网络文章、书籍、维基百科等多种来源。模型通过大量数据的学习来捕捉语言模式和结构。

Transformer 是近年来大语言模型的主要架构,它使用注意力机制来处理序列数据。主要特点包括:

- 自注意力机制:每个词(或子词)在序列中的表示可以与序列中其他所有词进行交互。这使得模型能够捕捉长距离依赖关系。

- 多头注意力:通过多个注意力头,模型能够从不同的“视角”捕捉序列中的信息。

- 位置编码:由于 Transformer 不具备内在的序列顺序信息,位置编码用于保留输入序列的位置信息。

模型在大规模无标注数据上进行预训练,学习通用的语言表示。这通常通过语言建模任务(如掩码语言建模或因果语言建模)实现。

在预训练基础上，模型可以在特定任务（如文本分类、问答系统等）上进行微调。这使用较小的标注数据集，并根据具体任务调整模型参数。

大语言模型可以用于生成文本或进行各种语言理解任务。生成文本时，模型根据给定的上下文预测下一个词或子词，逐步生成完整句子或段落。在理解任务中，模型对输入文本进行编码并生成相应的输出，如分类标签或答案。

大语言模型在多个领域中具有广泛应用，包括但不限于：

- 文本生成：如新闻摘要、故事生成、代码生成等。
- 语言翻译：自动将一种语言翻译为另一种语言。
- 问答系统：如智能助手和客服机器人。
- 文本分类：如情感分析、垃圾邮件检测等。

通过不断的研究和优化，大语言模型在准确性和效率上不断提高，推动了 NLP 领域的快速发展。

大语言模型作为现代自然语言处理的重要工具，依靠其强大的学习能力和灵活性，已经在各个领域展现出巨大潜力。未来，随着技术的进一步发展，语言模型将会变得更加智能和高效，继续为人类社会带来深远影响。

5.1.2 大语言模型的局限性

大语言模型在自然语言处理领域展现了巨大潜力，但也存在一些显著的局限性。这些局限性可以归纳为以下几类：

- 数据依赖性：训练数据集中的偏见会导致模型输出具有类似偏见且大规模数据收集可能涉及用户隐私问题，模型可能无意中泄露训练数据中的敏感信息。
- 计算资源：训练和运行大规模模型需要大量的计算时间和能源，成本高昂。大量计算资源的使用伴随着高能耗，对环境产生不利影响。
- 可解释性：神经网络模型的复杂性使其内部工作机制难以理解，决策透明度低。模型复杂性增加了发现和修正错误或偏差的难度。
- 通用性与专业性：在需要专业知识的领域，模型可能表现欠佳。微调模型以适应特定任务的效果可能不如专门设计的模型。
- 安全性与伦理：模型生成的高质量文本可能被恶意用于生成虚假信息、垃圾邮件或其他恶意内容。使用模型可能引发伦理问题，如生成的内容具有误导性或冒犯性，甚至被用来操纵舆论。
- 一致性与稳定性：模型有时生成不一致、不准确或缺乏连贯性的内容。模型输出的稳定性

可能受到输入微小变化的影响，导致不期望的结果。

5.1.3 大语言模型的应用场景总结

大语言模型在自然语言处理领域具有广泛的应用，主要包括以下几个方面：

- 文本生成：自动生成新闻摘要，帮助快速了解新闻核心内容；根据提示生成完整故事，应用于文学创作和娱乐；生成代码段，辅助编程，提高开发效率。

- 语言翻译：将一种语言自动翻译为另一种，促进跨语言交流；结合语音识别，实现实时语音翻译，应用于国际会议、旅游等场景。

- 问答系统：理解用户查询并提供准确回答，应用于智能助手如 Siri、Alexa；处理常见问题，提供即时帮助，减轻人工客服负担。

- 文本分类：分析文本情感倾向，应用于市场分析、社交媒体监控等；识别和过滤垃圾邮件，提高电子邮件系统的安全性。

- 信息检索：理解用户查询意图，提供更准确搜索结果；帮助快速找到相关信息，应用于法律、医学等领域。

5.1.4 论文搜索的方法

传统方法搜索引擎：使用 Google Scholar、Microsoft Academic 等工具查找论文。论文数据库：如 IEEE Xplore、PubMed 等，提供丰富的学术资源。大模型结合搜索引擎的方法步骤：

- 提供关键词和搜索条件。
- 大模型通过搜索引擎查询相关内容。
- 整理和总结搜索结果。

值得注意的是，一些平台允许大模型访问外部论文数据库，可以结合查询和分析功能进行深入研究。并且，在浏览的时候采用无痕浏览模式可能可以避免记录使用次数，可以免费使用更多功能。

在使用大模型进行搜索的时候可以通过以下几种方式进行更加精确的搜索：

- 根据核心论文拓展相关文献。
- 通过关键词、热门标题或 DOI 进行搜索。
- 提供 PDF 下载和相关文献的引用信息。
- 根据引用数量和时间排序相关论文，帮助用户快速找到重要研究。

5.2 大模型论文阅读辅助

在当今信息爆炸的时代，学术研究人员面临着海量文献资料的挑战。有效地获取、阅读并理解这些资料，是推动科学进步的关键。尤其在人工智能领域，大模型的出现极大地改变了信息处理的方式。这些模型不仅能够生成文本、解答问题，还能够辅助研究人员高效地阅读和理解复杂的学术论文。

本小节旨在探讨大模型如何辅助学术论文的阅读。第一部分将深入讲解 Retrieval-Augmented Generation (RAG) 的原理，这是一种结合了信息检索与生成的技术，能够通过查询大规模知识库来增强答案的准确性和深度。第二部分将介绍如何使用 ChatDoc 进行论文阅读辅助，展示其在处理学术文本时的实用性和效率。第三部分则探索 Kimi 模型的使用，这是一种专门为文献回顾和信息提取设计的工具，能够帮助研究人员快速定位关键信息并解析复杂数据。

通过这三部分的讨论，本小节希望为学术界提供一个全面的视角，理解和利用这些前沿技术来优化他们的研究流程。

5.2.1 Retrieval-Augmented Generation (RAG) 相关介绍

在人工智能的迅猛发展中，自然语言处理 (NLP) 技术在学术和工业界都有广泛应用。传统的语言模型虽然能够生成流畅的文本，但在处理需要深入专业知识的复杂查询时往往力不从心。这一局限性促使了结合检索和生成的新型模型——Retrieval-Augmented Generation (RAG) 的诞生。RAG 是一种创新的 NLP 框架，它结合了强大的检索功能和先进的生成技术，目的是提高模型处理复杂信息任务的能力，特别是在需要准确回答基于广泛知识库的问题时。

检索阶段：在 RAG 模型中，检索阶段首先通过一个索引化的大规模数据库（如 Wikipedia 或专业学术库）来处理输入问题。使用向量搜索技术，模型快速定位与问题最相关的文档或信

息片段。这一步是模型能够提供准确信息的关键。

生成阶段：获取到相关信息后，RAG 的生成组件（通常是一个基于 Transformer 的模型）将这些信息与原始查询一同处理，生成详细且准确的答案。这一过程中，模型不仅利用了从检索系统获得的上下文，还优化了信息的表达方式，使得输出更加自然和专业。

反馈机制：RAG 模型还包含一个反馈机制，通过用户互动或后续任务的性能反馈来不断调整检索和生成的策略。这一机制确保了模型在实际应用中的持续学习和优化。

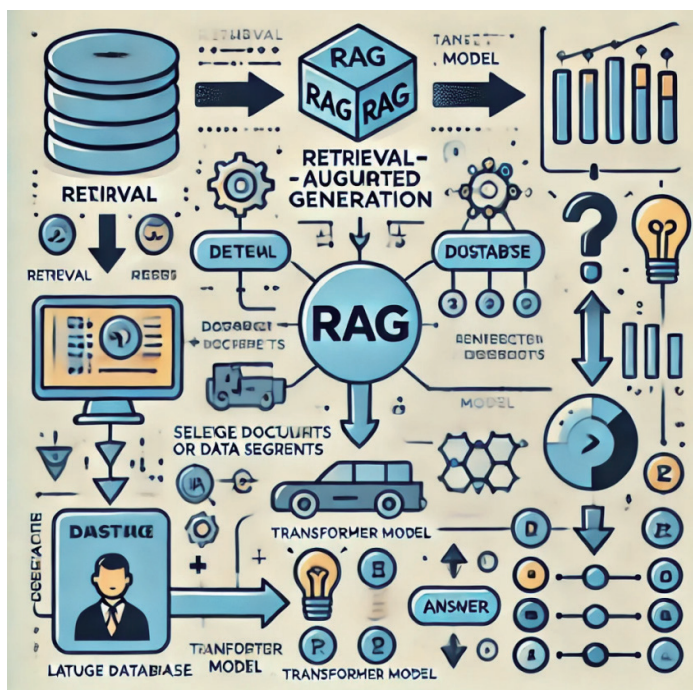


图 5.1: RAG 工作原理流程图

RAG 在学术阅读中的应用，一是提高理解深度：在学术研究中，RAG 可以通过提供与研究问题直接相关的文献背景，帮助研究人员更深入地理解特定的科学问题或理论。这对于处理跨学科的复杂问题尤其有价值；二是提升信息检索效率：RAG 通过自动化的高效检索减少了文献搜索的时间，允许研究人员更快地聚焦于分析和创新工作，而不是耗费大量时间在信息检索上。

通过结合强大的检索能力和先进的文本生成技术，RAG 为处理复杂的语言处理任务提供了一种有效的解决方案。它在学术领域尤其有潜力，可以极大地提高研究效率和质量。随着技术的不断进步，RAG 有望在学术和其他领域发挥更大的作用。

5.2.2 ChatDoc 使用教程

ChatDoc 是一个基于人工智能的工具，专门设计来帮助用户更高效地阅读和理解学术文献。

它结合了自然语言处理技术和机器学习算法，能够自动提取文档中的关键信息，生成摘要，并回答与文档内容相关的查询。ChatDoc 官网地址：https://chatdoc.com/chatdoc/#/upload。

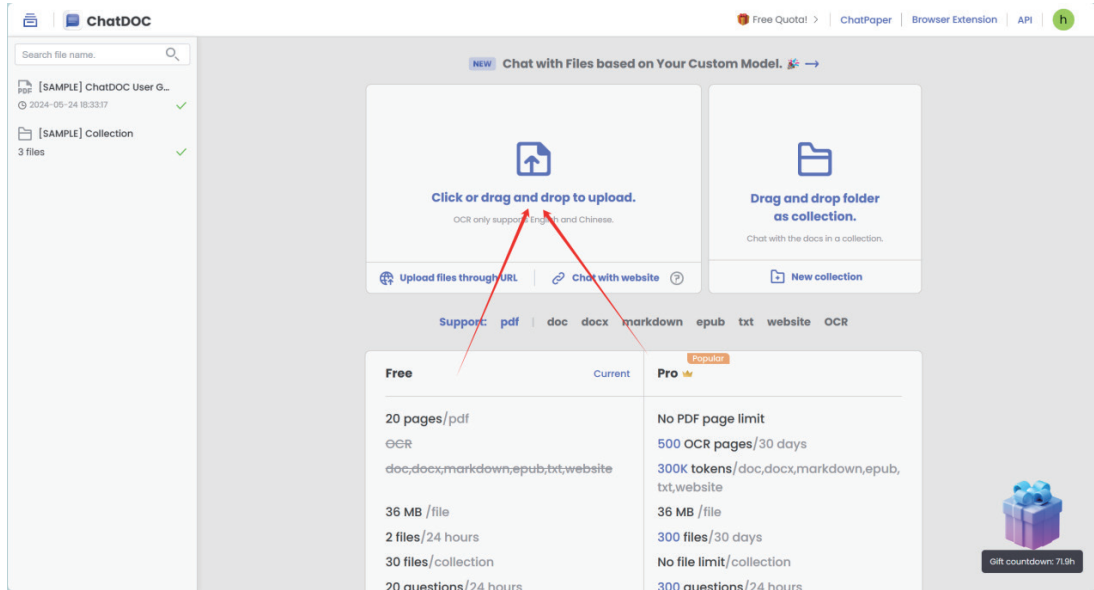


图 5.2: ChatDoc 官网及文件上传路径

ChatDoc 使用教程：首先，需要通过魔法访问 ChatDoc 官网并上传需要阅读的论文；其次，论文上传成功和解析完成后即可对论文内容进行提问；一方面，可以对整篇论文提问，如询问论文主旨大意，ChatDoc 会将答案以及参考段落一并给出，方便验证答案的正确性以及抓住论文要点；另一方面，也可以对单一段落进行提问，点击段落旁的蓝色小图标即可选中段落，如对选中的段落进行翻译和解释。

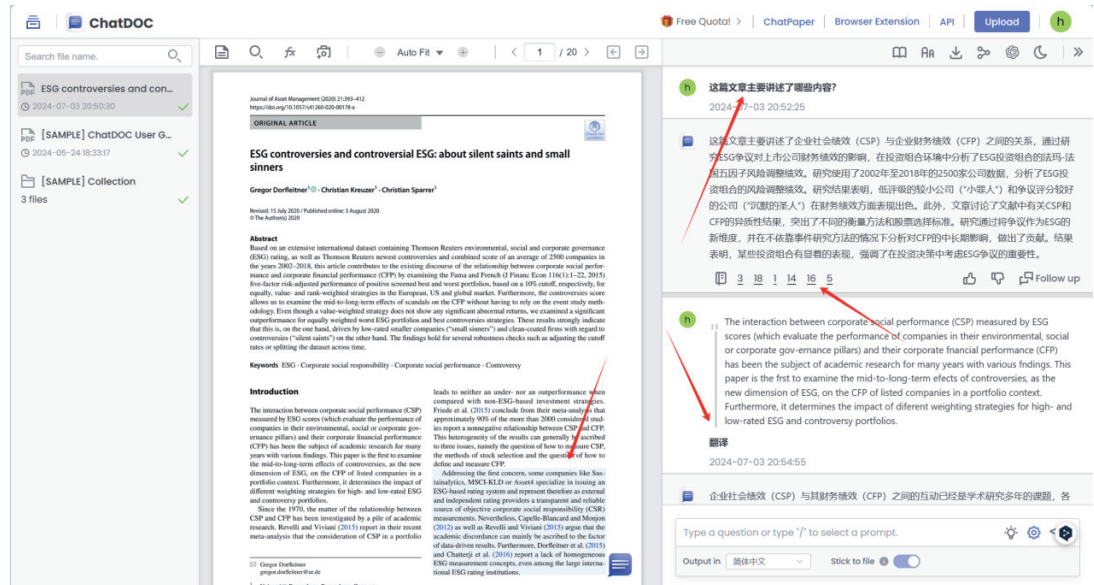


图 5.3: ChatDoc 论文阅读辅助功能

ChatDoc 的优点是内置大模型为 ChatGPT-3.5，回答结果更为详细和准确，同时，论文内容在左侧，回答结果在右侧，方便对照，极大程度上提高工作效率。ChatDoc 的缺点有两个，第一个是需要魔法才能访问官网，第二个是如果不付费的话，使用条件有所限制，如只能上传 20 页以内的论文、24 小时内只能询问 20 条问题。

5.2.3 Kimi 使用教程

Kimi 是一个专为学术和研究环境设计的人工智能工具，旨在帮助用户高效地管理、搜索和分析大量的学术文献。这个工具使用先进的自然语言处理技术来解析文本，提供深度学习算法支持的文献摘要，以及精确的信息检索功能。Kimi 官网地址：<https://kimi.moonshot.cn/>。



图 5.4: Kimi 官网及文件上传路径

Kimi 使用教程：首先，需要访问 Kimi 官网并上传需要阅读的论文；其次，论文上传成功和解析完成后即可对论文内容进行提问；一方面，可以对整篇论文提问，如询问论文主旨大意，kimi 会将答案直接给出，Kimi 最大输出字数为 10 万字，因此相较于 ChatDoc 的答案而言，Kimi 的答案更为详细；另一方面，也可以对单一段落进行提问，不需要选中段落，直接对某一段落提问即可，如对第一段落进行翻译和解释。



图 5.5: Kimi 论文阅读辅助功能

Kimi 的优点有很多，如不需要魔法即可访问官网提问、不需要收费即可对页数较多的论文进行提问且提问不限次数、同时可以上传 50 个以内的文件进行综合提问等等优点。Kimi 的缺点是不能将论文原文与答案进行对照，这一点可能会降低阅读效率。

本小节详细探讨了 Retrieval-Augmented Generation (RAG) 原理以及 ChatDoc 和 Kimi 两种工具在论文阅读辅助中的应用。RAG 结合检索与生成技术，通过查询大规模知识库来增强答案的准确性和深度，极大提高了学术文献的理解深度和信息检索效率。ChatDoc 利用自然语言处理技术自动提取文档中的关键信息，生成摘要并回答相关查询，尽管需要付费和访问限制，但其详细和准确的回答极大地提高了工作效率。Kimi 则以其强大的文献管理、搜索和分析能力，不需要付费且访问方便，能够提供更为详细的答案，尽管缺乏对照功能但依然是高效的论文阅读辅助工具。通过利用这些前沿技术，研究人员可以优化他们的研究流程，更加专注于分析和创新工作，从而推动科学的进步。

5.3 大模型论文书写辅助

随着人工智能技术的突飞猛进，大模型（如 GPT 系列、BERT 等）正以前所未有的速度渗透到各行各业中。尤其在科研领域，这些具备强大数据处理能力和深度学习能力的“智慧大脑”，正在悄然改变着传统的论文书写模式。它们不仅能够协助研究人员快速准确地获取并解读海量学术信息，更能以独特的视角和见解，参与到选题、摘要撰写、前言构建、技术方法阐述乃至论文评审等各个环节，为科研工作注入新的活力与创造力。

Nature 的著名审稿人 Dritjon Gruda 说，“如果您负责任地使用，生成式人工智能（AI）在写作、编辑和同行评审中可以成为宝贵的助手。”他在文章《Three ways ChatGPT helps me in my academic writing》中这样说道，“我从生成式 AI 中获得的价值不是来自技术本身盲目地产生文本，而是来自与工具的互动，并使用我自己的专业知识来完善它产生的结果。我与聊天机器人之间的对话既增强了我的工作的连贯性，也随着时间的推移教会了我如何以更简单的方式描述复杂主题。”他几乎每天都在使用这些工具来完善他撰写的论文中的措辞，AI 甚至帮助他完善了这篇文章。

CAREER COLUMN | 08 April 2024

Three ways ChatGPT helps me in my academic writing

Generative AI can be a valuable aid in writing, editing and peer review - if you use it responsibly, says Dritjon Gruda.

By [Dritjon Gruda](#)



For Dritjon Gruda, artificial-intelligence chatbots have been a huge help in scientific writing and peer review. Credit: Vladimira Stavreva-Gruda

<https://www.nature.com/articles/d41586-024-01042-3>

1/6

图 5.6: Nature 著名审稿人 Dritjon Gruda

在这个充满变革的时代里，大模型无疑成为了科研道路上的一股重要推动力。它们以精准的数据挖掘、高效的文本生成和个性化的推荐策略，助力科研人员实现更高层次的知识创新与传播。而诸如 GPT-4 这样的审稿专家级选手，则更为我们展示了大模型在学术评价体系中的巨大潜能。

接下来，我们将围绕大模型在论文书写过程中的具体应用展开探讨，并重点介绍几款备受瞩目的科研辅助工具。这些工具各具特色，将在文献管理、摘要自动化生成、团队协作等方面展现非凡实力，协力促进科研活动的数字化转型。让我们携手步入这场由大模型引领的科研革新，共同见证科技力量如何改写人类认知边界，开启学术研究的新纪元。

5.3.1 利用 AI 辅助“人工智能”论文写作

智能选题

在学术探索的旅程中，选题环节犹如指南针，指引着研究方向和深度。一个出色的选题不仅需要具备价值性、创新性、可行性，还应与社会需求紧密相关，并具有前瞻性。ChatGPT 作为智能助手，能够提供信息检索、选题建议、思路探讨以及写作辅助，助力研究者精准定位并深入挖掘研究主题。

目标：确定一个具有研究价值和创新性的论文选题。操作步骤：

1. 描述您的专业背景和研究方向。

2. 提供您感兴趣的技术或理论领域。

3. 指定您希望获得的研究选题数量。

提示词公式：“专业背景 + 研究方向 + 感兴趣的技术或理论领域 + 期望获得的研究选题数量”

示例：“作为一名计算机科学专业的研究人员，我的研究方向是人工智能和机器学习。我对深度学习模型在自然语言处理（NLP）领域的应用特别感兴趣，尤其是如何利用这些模型来生成高质量的文本摘要。我期望获得 5 个具有研究价值和创新的论文选题。”

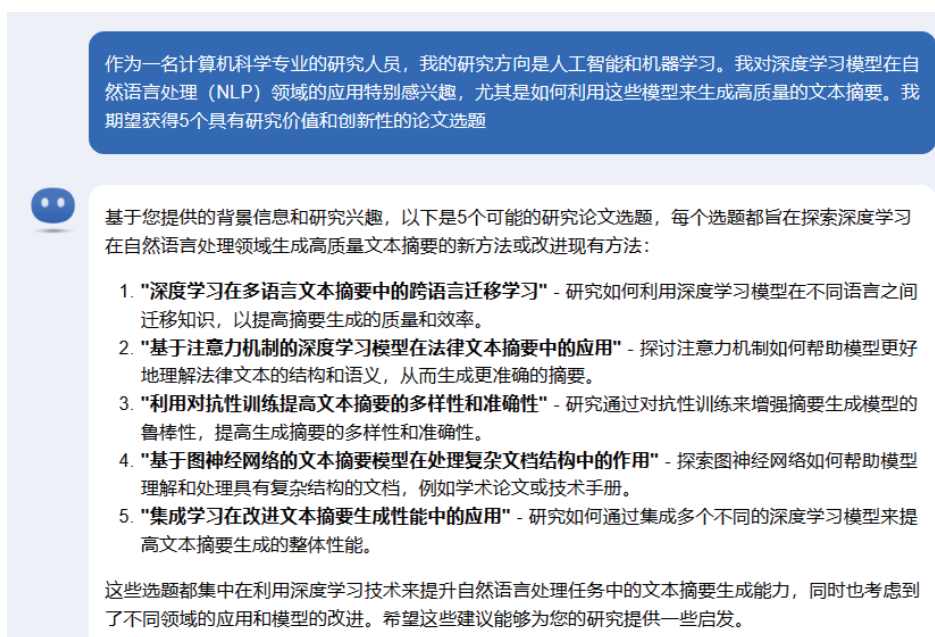


图 5.7：选题示例

摘要撰写

摘要是论文的橱窗，其重要性不言而喻。AI 大模型以其卓越的文本生成与优化能力，能够迅速捕捉论文核心，生成精炼而全面的摘要初稿。研究者可在 AI 生成的基础上进行个性化调整，进一步打磨摘要。AI 的语法、语义与逻辑分析能力，确保了摘要的准确性与流畅性。在金字塔原理指导下，AI 帮助研究者构建清晰的摘要结构——背景、冲突与解答，使读者能够迅速把握论文的主旨。

摘要生成步骤：

1. 揭示背景：概述研究领域的现状与重要性。
2. 引出冲突：点明研究面临的挑战或未解之谜。
3. 提供解答：阐明研究目标与创新解决方案。

摘要生成

步骤 1：了解背景

首先，我们需要理解论文的主题和主要内容。假设这篇论文讨论的是“人工智能在医疗诊断中的应用”，那么背景可以描述为：近年来，随着人工智能技术的发展，越来越多的医疗机构开始尝试将其应用于医疗诊断领域，以提高诊断精度和效率。

步骤 2：提出冲突 / 疑问

接着，我们需要找出研究的核心问题或冲突。例如：然而，现有的 AI 诊断系统仍存在误诊率较高、解释性差等问题，限制了其在临床上的应用。

步骤 3：给出解答

最后，我们要概述研究的目标和解决方案：本文旨在开发一种新型的 AI 诊断模型，通过融合深度学习和专家知识，降低误诊率并提高可解释性。

摘要精炼

将以上三部分合并成一段话，得到的摘要如下：

近年来，尽管人工智能在医疗诊断领域的应用越来越普遍，但误诊率高和解释性差的问题制约了其临床推广。为此，本文提出了一种融合深度学习和专家知识的新型 AI 诊断模型，旨在解决这些问题并提高诊断精度和透明度。

请根据金字塔原理 1. 背景：简要介绍研究的背景和意义。

2. 冲突/疑问：提出研究面临的问题或存在的矛盾。

3. 解答：阐述研究的目的是解决方案。撰写人工智能在医疗诊断中的应用摘要，先分别生成，后合并



背景：

随着人工智能技术的快速发展，其在医疗领域的应用正逐渐改变传统的诊断流程。人工智能，尤其是深度学习技术，已经在图像识别、模式识别和数据分析等方面展现出巨大的潜力。在医疗诊断中，AI的应用不仅可以提高诊断的准确性，还能加快诊断速度，降低成本，从而为患者提供更及时、更有效的医疗服务。

冲突/疑问：

尽管人工智能在医疗诊断中具有显著的优势，但其应用仍面临一些挑战。其中包括数据隐私和安全性问题、模型的泛化能力、以及医疗专业人员对AI技术的接受度。此外，如何确保AI诊断系统的决策过程透明且可解释，也是当前研究中需要解决的关键问题。

解答：

本研究旨在探索人工智能在医疗诊断中的具体应用，并提出解决方案以克服现有挑战。研究将集中在开发更为精确和可靠的AI诊断模型，同时确保这些模型的决策过程是透明的。此外，研究还将探讨如何通过教育和政策制定来提高医疗专业人员对AI技术的信任和接受度，以及如何制定相应的数据保护措施来确保患者隐私。

合并摘要：

在本研究中，我们首先介绍了人工智能在医疗诊断领域应用的背景和意义，强调了其在提高诊断准确性和效率方面的潜力。然后，我们指出了当前AI应用在医疗诊断中所面临的主要问题，包括数据隐私、模型泛化、专业人员接受度以及决策透明度。针对这些问题，本研究的目的是开发出更精确、可靠的AI诊断模型，并确保其决策过程的透明度。同时，本研究还将探讨如何通过教育和政策制定来提升医疗专业人员对AI技术的信任，以及如何制定有效的数据保护措施来保护患者隐私。通过这些研究活动，我们期望为人工智能在医疗诊断中的广泛应用提供坚实的理论和实践基础。

图 5.8：摘要生成示例

这段摘要遵循了金字塔原理的结构，清晰地呈现了研究的背景、问题和解决方案，便于读者迅速把握文章主旨。

前言构建

前言部分是论文的引路人，为读者提供研究背景、动机和目的等关键信息。AI 大模型通过分析论文主题和背景知识，生成逻辑性和连贯性强的前言，为读者揭开研究的序幕。

不含明确指示的提示词

我的学术论文题目是《综合经验与机器学习建模方法用于心脏病预测》，请撰写前言部分。



图 5.9：论文前言（不含明确的指示）

含明确指示的提示词

我的学术论文题目是《综合经验与机器学习建模方法用于心脏病预测》，请帮我撰写前言部分。要求：

1. 前言需交代研究背景、研究不足、研究价值和研究意义

2. 字数不少于 1000 字。

我的学术论文题目是《综合经验与机器学习建模方法用于心脏病预测》，请帮我撰写前言部分。要求：

1. 前言需交代研究背景、研究不足、研究价值和研究意义
2. 字数不少于 1000 字。



前言

心脏病作为全球范围内的主要死亡原因之一，对人类健康构成了巨大威胁。据世界卫生组织（WHO）统计，每年约有1700万人死于心血管疾病，其中心脏病占据了相当大的比例。心脏病的早期诊断和预防对于降低死亡率、提高患者生活质量具有重要意义。随着医学技术的发展，心脏病的诊断方法也在不断进步，从传统的心电图、超声心动图到现代的基因检测、生物标志物分析，各种技术手段都在为心脏病的诊断提供支持。然而，由于心脏病的复杂性和多变性，单一的诊断方法往往难以满足临床需求，因此，综合多种诊断方法和手段，利用机器学习建模技术进行心脏病预测，成为了医学研究的热点。

研究背景

心脏病的发病机制复杂，涉及遗传、环境、生活方式等多种因素。传统的诊断方法虽然在一定程度上能够为心脏病的诊断提供依据，但往往存在一定的局限性。例如，心电图虽然能够反映心脏的电生理活动，但对于某些类型的心脏病，如心肌梗死，其敏感性和特异性并不高。此外，传统的诊断方法往往依赖于医生的经验和主观判断，这在一定程度上影响了诊断的准确性和一致性。随着计算机技术的发展，机器学习作为一种高效的数据处理技术，为心脏病的诊断和预测提供了新的思路。

机器学习建模方法通过分析大量的临床数据，能够发现心脏病发病的潜在规律和模式，从而提高诊断的准确性。然而，目前机器学习在心脏病预测领域的应用仍处于起步阶段，存在诸多不足。首先，心脏病数据的收集和整理存在困难，数据的质量和完整性直接影响模型的预测效果。其次，现有的机器学习模型往往依赖于特定的数据集，缺乏泛化能力，难以适应不同的临床环境。此外，机器学习模型的解释性较差，难以为临床医生提供直观的诊断依据。

研究不足

尽管机器学习在心脏病预测领域展现出了巨大的潜力，但目前的研究仍存在一些不足。首先，大多数研究集中在模型的构建和优化上，而忽视了模型的临床应用和验证。其次，现有的研究往往缺乏对心脏病发病机制的深入探讨，导致模型的预测结果难以与病理生理过程相联系。此外，机器学习模型的可解释性问题也是当前研究的一大挑战，如何让模型的预测结果更容易被临床医生理解和接受，是亟待解决的问题。

研究价值

本研究旨在通过综合经验医学和机器学习建模方法，探索心脏病的预测模型，以期提高心脏病的早期诊断率和治疗成功率。研究的价值主要体现在以下几个方面：

1. 提高诊断准确性：通过机器学习模型的辅助，可以提高心脏病的诊断准确性，减少误诊和漏诊的发生。
2. 促进个性化治疗：机器学习模型可以根据患者的具体情况，提供个性化的治疗方案，提高治疗效果。
3. 优化资源配置：通过预测心脏病的发病风险，可以合理分配医疗资源，提高医疗效率。
4. 推动医学研究：本研究的成果可以为心脏病的发病机制研究提供新的视角和方法，推动相关领域的科学研究。

研究意义

心脏病的预测和早期诊断对于降低死亡率、提高患者生活质量具有重要的意义。本研究的意义主要体现在以下几个方面：

1. 社会意义：随着心脏病患者数量的增加，心脏病的预防和治疗已经成为社会关注的焦点。本研究的成果有助于提高心脏病的早期诊断率，减少心脏病对社会和家庭的影响。
2. 医学意义：本研究将为心脏病的诊断和治疗提供新的工具和方法，推动心脏病学的发展。
3. 经济意义：通过提高心脏病的诊断准确性和治疗效果，可以降低医疗成本，减轻患者的经济负担。
4. 科技意义：本研究将推动机器学习技术在医学领域的应用，为人工智能在医疗健康领域的研究提供新的案例和经验。

综上所述，本研究将综合经验医学和机器学习建模方法，对心脏病的预测进行深入研究，以期为中心病的早期诊断和治疗提供科学依据，具有重要的理论价值和实践意义。

(注：以上内容为示例性文本，实际撰写时应根据具体研究内容和数据进行调整和补充。)

图 5.10: 论文前言 (含明确的指示)

技术方法

技术方法是论文的骨架，支撑着研究的严谨性与创新性。AI 大模型在此环节中发挥着至关重要的作用，它能够清晰解释算法原理，介绍算法的发明者和背景，比较不同算法的优势，并详细指导算法的实施步骤。无论是在机器学习、深度学习还是其他技术领域，AI 大模型均能提供专业的技术支持和深入的分析。

参考提示词：在科学论文中，我打算编写一段内容，并希望你提供以下风格的信息：首先，简洁地解释【xx】算法(一两句话)，以及它为何能够发挥作用。其次，介绍提出这个算法的人。再次，探讨其相较于其他算法的优势。最后，详细描述如何实施该算法，包括分步骤的操作指南。



SVD算法简介及作用原理

SVD (奇异值分解) 是一种线性代数中的矩阵分解技术，它将任意矩阵分解为三个矩阵的乘积，能够揭示矩阵内在的数据结构和特征。SVD之所以有效，在于它能以较少的奇异值和向量捕捉到原始数据集的主要信息，从而实现数据压缩、降噪和特征提取等目的。

提出者

SVD概念最早由Eugenio Beltrami于1873年提出，但其现代形式和广泛使用是在20世纪初，特别是在数学家Erhard Schmidt和Camille Jordan的工作中得到了发展和完善。不过，SVD作为一个实用的算法，其现代理解和应用很大程度上归功于计算机科学和数值分析领域的发展。

相较于其他算法的优势

SVD相比其他矩阵分解方法，如PCA (主成分分析)，具有更强的稳定性和普适性。它可以直接应用于非方阵，无需对数据进行中心化，而且SVD的奇异值具有直观的物理意义，代表了数据的不同维度上的重要性。此外，SVD能够有效地处理缺失值和噪声，是推荐系统、图像压缩、信号处理等多个领域中的首选工具。

实施SVD算法的步骤

1. **准备矩阵**: 首先，确保你的数据已经组织成一个矩阵形式，其中行代表观测样本，列代表特征。
2. **计算矩阵的转置与乘积**: 计算矩阵A的转置 A^T 与A的乘积 $A^T A$ ，以及A与A的转置的乘积 $A A^T$ 。
3. **求解特征值和特征向量**:
 - 对于 $A^T A$ ，求解特征值 λ_i 及其对应的特征向量 u_i 。
 - 对于 $A A^T$ ，求解特征值 σ_i^2 及其对应的特征向量 v_i 。
4. **构建 Σ 矩阵**: 根据求得特征值 σ_i ，构建对角矩阵 Σ ，其中对角线元素为 σ_i 的平方根，即奇异值。
5. **标准化特征向量**: 将求得特征向量 u_i 和 v_i 单位化，得到左奇异向量U和右奇异向量V。

6. **重构矩阵**: 最终的SVD分解结果为 $A = U\Sigma V^T$ 。通过上述步骤, 原始矩阵A被分解为 $U\Sigma V^T$ 的形式, 其中U和V是正交矩阵, Σ 是对角矩阵, 包含了矩阵A的奇异值。SVD的这一特性使得它在数据分析和处理中极其强大和灵活。

图 5.11: SVD 算法介绍

论文润色

在学术研究的世界里, 论文的撰写和润色是一个至关重要的环节。一篇精心打磨的论文不仅能够清晰地传达研究者的发现和观点, 还能在同行评审和学术交流中获得更高的认可。随着人工智能技术的飞速发展, AI 辅助的论文润色工具已经成为学术写作的有力助手。这些工具通过先进的算法和自然语言处理技术, 为研究者提供了前所未有的便利和效率。

AI 润色工具的核心优势在于它们能够快速识别并纠正文本中的语法错误、拼写错误以及风格上的不一致性。此外, 它们还能够根据学术写作的规范和特定期刊的要求, 提供定制化的润色建议。这不仅大大节省了研究者的时间, 也提高了论文的专业性和可读性。

在 AI 辅助论文润色的过程中上下文是至关重要的。我们不能指望生成式 AI 在没有任何信息的情况下对一个问题给出有意义的回应。所以当您使用聊天机器人来为您的论文的一个部分修饰时, 首先要对论文的背景给出一些信息, 并且针对不同的问题可以使用不同的 AI 工具。Dritjon Gruda 说, 他通常使用由加利福尼亚州旧金山的 OpenAI 制作的 ChatGPT, 但对于需要深刻理解语言细微差别的任务, 例如分析搜索查询或文本, 他发现由谷歌研究人员开发的 Gemini 特别有效。当他离线工作但仍需要聊天机器人的帮助时, Mixtral 在巴黎制作的开源大型语言模型是理想的选择。

但无论您选择哪种生成式 AI 工具, 成功的关键都在于提供精确的指令, 您的描述越清晰, 就越有可能得到更好的效果。当您使用聊天机器人来完善您论文的某一部分以提高清晰度时, 首先要概述上下文: 您的论文是关于什么的, 您的主要论点是什么, 以什么样的格式呈现等等。例如, 你可能会这样写: “我正在为领先的 [学科] 学术期刊撰写一篇关于 [主题] 的论文。我试图在以下部分说的是 [特定点]。请为了清晰度、连贯性和简洁性重新表述它, 确保每个段落都能自然地过渡。去掉行话, 使用专业的语调。” 之后您依然可以再次使用相同的技术来完善您的表述。

大多数时候聊天机器人的第一次回复可能不是完美的——它是一个协作和迭代的过程。您可能需要完善您的指令或添加更多信息, 就像您与同事讨论一个概念时一样。正是这种互动提高了结果。如果某些东西不太符合要求, 不要犹豫说, “这不太符合我的意思。让我们调整这部分。” 或者您可以称赞它的改进: “这更清晰了, 但让我们调整结尾, 以便更强烈地过渡到下一节。”

这种方法可以将一个具有挑战性的任务转变为一个可管理的任务，填满你可能没有完全领会的洞察力。这就像进行了一次打开新视角的对话，使生成式 AI 成为发展和完善想法的创造性协作伙伴。

在接下来的内容中，我们将深入探讨如何利用 AI 进行论文润色，并提供一系列实用的提示词，以帮助研究者在实战中提升写作质量。

中文论文优化提示词

1. 作为一名中文学术论文写作改进助理，你的任务是改进所提供文本的拼写、语法、清晰、简洁和整体可读性，同时分解长句，减少重复，并提供改进建议。请只提供文本的更正版本避免包括解释。请编辑以下文本【我们的方法叫作掩膜 R-CNN，通过添加用于每个感兴趣区域 (RoI) 的掩膜分割预测并与用于分类和边界框回归分析的现有分支并行的分支，它拓展了极速 R-CNN[34]。】

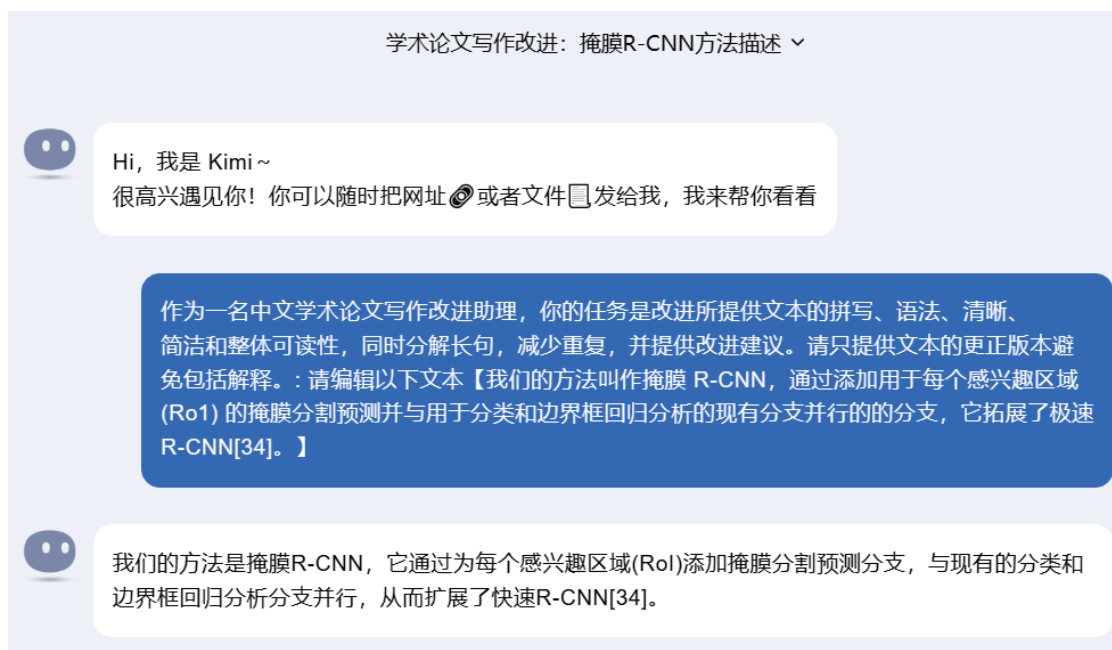


图 5.12: kimi 中文论文优化示例 1

2. 你现在扮演一个【人工智能】领域的专家，从专业的角度，您认为下面这些内容是否需要优化的地方？注意，不要全文修改，您需要 – 指出需要修改的地方，并且给出修改意见以及推荐的修改内容。要润色的段落是【我们的方法叫作掩膜 R-CNN，通过添加用于每个感兴趣区域 (RoI) 的掩膜分割预测并与用于分类和边界框回归分析的现有分支并行的分支它拓展了极速 R-CNN[34]。】

你现在扮演一个【人工智能】领域的专家，从专业的角度，您认为下面这些内容是否需要优化的地方？注意，不要全文修改，您需要一指出需要修改的地方，并且给出修改意见以及推荐的修改内容。要润色的段落是：【我们的方法叫作掩膜 R-CNN，通过添加用于每个感兴趣区域 (Ro1) 的掩膜分割预测并与用于分类和边界框回归分析的现有分支并行的分支它拓展了极速 R-CNN[34]。】



需要修改的地方及修改意见如下：

1. **错误拼写**: "Ro1" 应更正为 "RoI" (Regions of Interest的缩写)。
2. **语法冗余**: 句子中的 "的的" 重复，应删除一个以避免重复。
3. **语法结构**: 句子结构较为复杂，可以分解为更简洁的句子，以提高清晰度和可读性。
4. **术语准确性**: "极速 R-CNN" 应使用正确的术语，如果是文献中已有的名称，应保持一致性。

推荐修改内容如下：

- 将 "Ro1" 修改为 "RoI"。
- 删除重复的 "的" 字。
- 将复杂句子拆分为两个或更多的简单句子，以提高可读性。
- 确认 "极速 R-CNN" 是否为文献中的标准术语，如果不是，应使用正确的术语名称。

修改后的句子可能如下：

我们的方法称为掩膜 R-CNN，它通过为每个感兴趣区域(RoI)添加掩膜分割预测，与现有的分类和边界框回归分析并行工作。这种方法扩展了快速 R-CNN[34]。

图 5.13: kimi 中文论文优化示例 2

论文润色及原因分析提示词

以下是一篇学术论文的段落。润色写作以符合学术风格，提高拼写、语法、清晰度、简洁性和整体可读性。必要时，重写整个句子。【我们的方法叫作掩膜 R-CNN，通过添加用于每个感兴趣区域 (Ro1) 的掩膜分割预测并与用于分类和边界框回归分析的现有分支并行的分支，它拓展了极速 R-CNN[34]。在 Markdown 表格中列出所有修改并解释这样做的原因，第 1 列是原文，第 2 列是修改后的句子，第 3 列是修改原因。

以下是一篇学术论文的段落。润色写作以符合学术风格，提高拼写、语法、清晰度、简洁性和整体可读性。必要时，重写整个句子。【我们的方法叫作掩膜R-CNN，通过添加用于每个感兴趣区域(Ro1)的掩膜分割预测并与用于分类和边界框回归分析的现有分支并行的分支，它拓展了极速R-CNN[34]。在 Markdown 表格中列出所有修改并解释这样做的原因，第1列是原文，第2列是修改后的句子，第3列是修改原因。



原文	修改后的句子	修改原因
我们的方法叫作掩膜R-CNN	我们提出的方法，称为掩膜R-CNN	增加"提出"二字，以明确方法的创新性

通过添加用于每个感兴趣区域(RoI)的掩膜分割预测	通过为每个感兴趣区域(RoI)引入掩膜分割预测	修正了拼写错误, 使用"引入"替换"添加", 以提高语言的学术性
并与用于分类和边界框回归分析的现有分支并行的分支	同时与现有的分类和边界框回归分析分支并行工作	删除了重复的"的", 简化了句子结构, 提高了清晰度
它拓展了极速R-CNN[34]	从而扩展了快速R-CNN框架[34]	修正了术语错误, 使用"快速R-CNN"替换"极速R-CNN", 并明确指出是对框架的扩展

修改后的段落如下:

我们提出的方法, 称为掩膜R-CNN, 通过为每个感兴趣区域(RoI)引入掩膜分割预测, 同时与现有的分类和边界框回归分析分支并行工作, 从而扩展了快速R-CNN框架[34]。

图 5.14: 列表分析示例

论文翻译和润色提示词

请你充当【xxx】领域论文编辑专家, 将汉语翻译为英文, 修正语法错误, 让段落表达更学术化, 逻辑更清晰并多采用 therefore, however, hence, consequently, although 等志逻辑关系的转折词。注意, 我不需要你在原文上扩展, 只需要润色我给出的段落。要润色的段落是:【自然语言的处理任务, 比如 question answering, machine translation, reading comprehension, and summarization, 通常使用特定的 supervised learning 在特定任务中, 并且依赖于 specific datasets.】

请你充当【nlp】领域论文编辑专家, 将汉语翻译为英文, 修正语法错误, 让段落表达更学术化, 逻辑更清晰并多采用 therefore, however, hence, consequently, although 等志逻辑关系的转折词。注意, 我不需要你在原文上扩展, 只需要润色我给出的段落。要润色的段落是:【自然语言的处理任务, 比如 question answering, machine translation, reading comprehension, and summarization, 通常使用特定的 supervised learning 在特定任务中, 并且依赖于 specific datasets.】



Original Paragraph:

自然语言的处理任务, 比如 question answering, machine translation, reading comprehension, and summarization, 通常使用特定的 supervised learning 在特定任务中, 并且依赖于 specific datasets.

Refined Paragraph:

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, typically employ specific instances of supervised learning for their respective tasks and are contingent upon specialized datasets.

Reasons for Changes:

1. "自然语言的处理任务" was translated to "Natural language processing tasks" to maintain the subject of the sentence in English.
2. "比如" was translated to "such as" to provide a list of examples in English.
3. The phrase "使用特定的 supervised learning" was translated to "employ specific instances of supervised learning" to clarify that different forms of supervised learning are used for different tasks.

4. "在特定任务中" was translated to "for their respective tasks" to indicate that each task has its own specific application of supervised learning.
5. "并且依赖于specific datasets" was corrected to "and are contingent upon specialized datasets" to use the correct word order and to convey the dependency on specific types of data more formally.
6. The sentence structure was adjusted to improve academic tone and clarity. Logical connectors were not added as the original sentence did not provide a context for their use.

图 5.15: 论文翻译和润色示例

GPT-4: 审稿专家中的佼佼者

近期, 斯坦福大学及其他研究机构开展的一项实验揭示了令人瞩目的事实: 当数千篇顶级期刊如《自然》(Nature) 和国际学习表征会议 (ICLR) 的文章被提交给 GPT-4 以获取审稿反馈时, 这款先进的人工智能模型不仅圆满完成了任务, 其表现甚至超越了许多人类审稿人。统计数据表明, GPT-4 所提出的审稿意见中, 超过 50% Nature 的著名审稿人 Dritjon Gruda 说: “生成式 AI 在同行评审过程中可以成为宝贵的工具。在全面阅读手稿后, 总结关键点和审查领域, 并使用 AI 来帮助组织和表达您的反馈” 此外, Dritjon Gruda 还说到, 他从聊天机器人中获益的最后一个领域是作为期刊编辑的角色。特别是当他每周都要监督几篇手稿时, 向作者提供建设性的编辑反馈可能很具挑战性。在这个过程中, ChatGPT 已成为了他不可或缺的工具, 帮助他制定精确、富有同情心和可操作的反馈, 而不取代人类编辑决策。

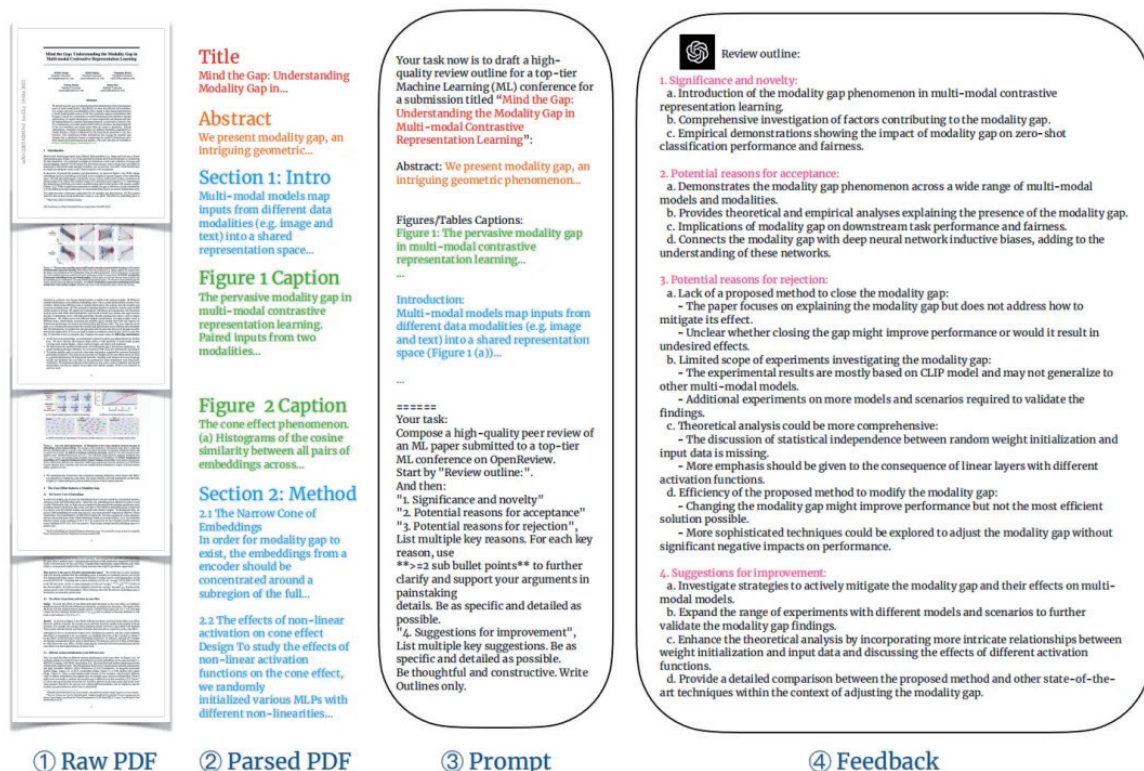
例如, 在评估一篇论文并指出其优缺点后, 他可能会将这些输入 ChatGPT 并让它起草一封合适的信件: “基于这些笔记, 起草一封给作者的信。突出手稿的关键问题, 并清楚地解释为什么手稿, 尽管主题有趣, 可能没有足够的进展来值得发表。避免使用行话, 直接了当, 始终保持专业和尊重的语调。” 这种方法既提高了反馈的质量, 也有助于确保以支持性的方式传达审稿人的想法。结果是编辑和作者之间更积极和富有成效的对话。

论文合著者 James Zou 总结道: “虽然我们依然需要高质量的人类反馈, 但 LLM 在正式的同行评审前, 能够协助作者有效改善论文初稿, 其作用不容小觑。”

想要体验 AI 审稿的魅力, 操作其实十分简便: 只需将论文的 PDF 文件转化为文本, 输入至 GPT-4, 即可迅速获得反馈。具体流程涉及从 PDF 文件中抽取论文标题、摘要、图表、表格标题及正文内容。随后, 向 GPT-4 明确指示, 要求其遵循顶尖学术期刊和会议的审稿标准, 涵盖四个方面: 成果的重要性和创新性评估、可能的接受理由、可能的拒绝理由以及具体的改进建议。例如, 您可能会指示 AI: “假设您是一位拥有 20 多年学术经验的专家和经验丰富的学者, 基于您对 [领域] 中一篇论文的总结, 其中主要关注 [一般主题], 请详细审查这篇论文, 按以下顺序: 1) 简要讨论其核心内容; 2) 识别其局限性; 3) 按重要性顺序解释每个局限性的重要性,

在整个过程中保持简洁和专业的语调。”

AI 合作可以非常丰富，这些工具可能为您提供未曾考虑过的视角。例如，ChatGPT 在解释和论证特定局限性背后的原因方面表现出色，如果您识别出某种方法论的局限性，ChatGPT 可以详细阐述这些局限性，并建议如何在修订中克服它们。这种反馈通常有助于将局限性及其对论文整体贡献的影响联系起来。但在某些情况下，它的建议可能偏离基础，牵强附会，不相关甚至完全错误，所以审稿人必须能够区分什么是事实，什么是虚构，这是聊天机器人不能可靠地做到的。



Supplementary Figure 5. Schematic of the LLM scientific feedback generation system. Manuscript text, including figure captions, is extracted from the manuscript PDFs and integrated into a prompt for LLM GPT-4, which then generates feedback. The generated feedback provides structured comments in four sections: significance and novelty, potential reasons for acceptance, potential reasons for rejection, and suggestions for improvement. In the example, GPT-4 raised a comment that the paper reported a modality gap phenomenon but did not propose methods to close the gap or demonstrate the benefits of doing so.

图 5.16: GPT 审稿技术路线

为了引导 GPT-4 生成结构清晰、内容详实的审稿意见，可以参考以下提示词模板：

您的任务：为提交至 OpenReview 平台、针对顶级会议的论文撰写高质量的同行评审报告。

请按照以下框架展开：

1. 重要性和创新性：明确指出论文解决的关键问题、创新点及其对领域发展的贡献。

2. 可能的接受理由：评价研究方法的有效性、结果的显著性和可重复性，以及文献回顾的全面性和引用的恰当性。

3. 可能的拒绝理由：列出四个关键理由，并为每个理由提供至少两个支持细节，涉及方法论局限、数据集代表性、结果解读偏差及讨论部分的充分性。

4. 改进建议：提出四条具体建议，涵盖实验设计的优化、增加对比实验的必要性、文章结构与语言表达的改进，以及对未来研究方向的展望。

写作指南：请以“Review outline:”开头，随后逐一论述上述要点。务必保持审稿意见的专业性与建设性，确保反馈内容既全面又深入，有助于作者理解和提升论文质量。

通过精心设计的提示词，我们不仅限定了审稿的结构框架，还明确了审稿人应扮演的客观公正角色，以及对审稿内容的专业水准和建设性反馈的要求。这将促使大模型生成的审稿意见更加贴近实际学术评审的高标准，为作者提供真正有价值、可操作的改进建议，促进论文质量的显著提升。

5.3.2 相关软件与工具

通过前文的案例分析，我们深刻认识到大型语言模型在辅助论文写作方面所展现出的卓越能力。它们在选题创意、摘要精炼、前言铺陈、技术路径规划，乃至论文审校等各个环节，均提供了智能化、个性化、自动化的高效支持，极大地优化了学术写作的流程和成果。展望未来，随着这些大型模型技术的不断演进，它们在学术研究领域的应用将变得更加广泛和深入，为科研人员开拓更广阔的便利与创新空间。

在此基础上，本书将继续深入探讨两款具体的辅助工具：SciSpace、Paper Digest 和星火科研助手。这些工具将进一步展示如何将人工智能技术融入学术研究的各个阶段，为科研工作提供更为全面和细致的辅助。

SciSpace

SciSpace 的发展历史可以追溯到其前身——Papers，这是一个由三名荷兰工程师在 2008 年创立的文献管理软件。Papers 最初的设计目的是为了帮助科研人员更好地组织和管理他们的学术文献，提供了一个直观的界面来存储、检索和引用文献。随着时间的推移，Papers 逐渐增加了更多功能，包括 PDF 阅读器、注释工具、引用管理和自动文献导入等，使其成为科研领域内颇受欢迎的文献管理解决方案之一。

2014 年，Papers 被 Mendeley 公司收购，后者是一家同样专注于科研文献管理的英国初创

企业。这次合并意在整合双方的优势，提供更全面的科研工作流解决方案。然而，好景不长，2015 年 Mendeley 又被全球最大的学术出版商爱思唯尔（Elsevier）收购。

在爱思唯尔的麾下，Papers 经历了一系列变革，包括品牌重塑和技术升级。2018 年，Papers 正式更名为 Mendeley Desktop，成为 Mendeley 产品线的一部分，继续提供文献管理服务。

但是，就在同年，SciSpace 作为一个独立的品牌和平台诞生了。它由原 Papers 的核心团队成员创立，旨在延续并超越 Papers 的愿景，利用最新的 AI 技术和云服务，打造一个更智能、更高效的科研文献管理和协作平台。SciSpace 不仅继承了 Papers 的文献管理功能，还引入了 AI 驱动的文献理解、问答和摘要提取等创新功能，使其在科研辅助工具领域独树一帜。

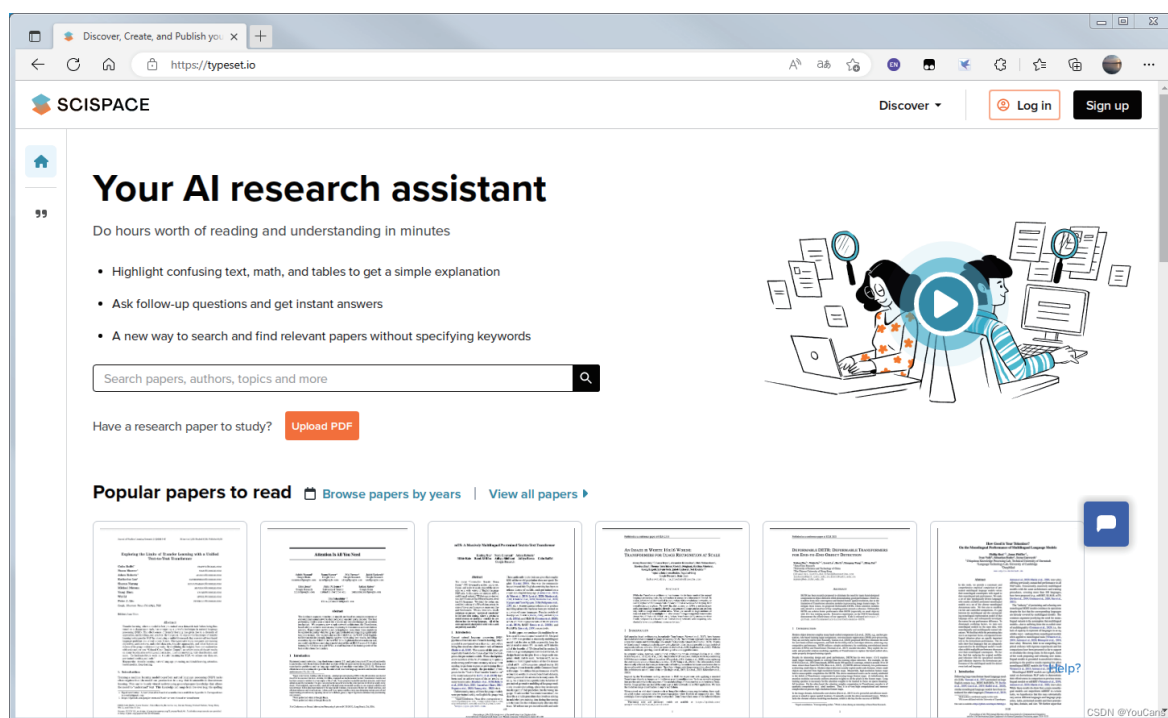


图 5.17: SciSpace 使用界面

自成立以来，SciSpace 持续迭代和优化其产品，致力于为科研人员提供更便捷、智能的文献处理体验。通过不断的技术革新和用户体验改进，SciSpace 正在逐步成为科研领域内的一股重要力量，助力科研人员高效地进行文献研究和学术交流。Paperdigest

Paper Digest

PaperDigest 的故事始于 2019 年前后，彼时，自然语言处理 (NLP) 与深度学习技术正蓬勃发展，科研人员面临着在海量文献中快速获取关键信息的挑战。正是在这种背景下，PaperDigest 应运而生，旨在利用先进的 AI 算法自动生成论文摘要，从而极大提升文献检索与阅读的效率。

起初，团队致力于探索如何精准捕捉论文的核心观点，同时保持摘要的简洁与连贯，这是

对 NLP 技术的一次重要考验。

2020 年，PaperDigest 首次公开亮相，在 TechCrunch Disrupt Startup Battlefield 上引起了广泛关注，其独特的价值主张和技术潜力获得了业界的初步认可。随后，2021 年，PaperDigest 通过种子轮融资获得了资本的支持，这不仅加速了技术的研发进程，也推动了产品功能的丰富与完善。团队深入优化 AI 算法，使得摘要的准确度与相关性显著提升，同时，引入了文献管理、关键词提取等实用特性，满足了不同用户群体的多样化需求。

到了 2022 年，PaperDigest 与 Springer Nature 等学术出版巨头达成合作，这不仅拓宽了其学术资源的边界，也为摘要的质量和覆盖范围带来了质的飞跃。更重要的是，这一阶段，PaperDigest 开始探索 AI 技术在学术写作中的更深层次应用，如智能论文撰写辅助、同行评审建议系统等，预示着其在科研工作流程中的角色将愈发重要。

进入 2023 年及未来，PaperDigest 依然保持着对技术创新的执着追求，致力于在 NLP 领域深耕细作，探索诸如语义分析、情感分析等前沿技术，旨在进一步提升摘要的生成质量，同时开拓更多应用场景，如自动论文初稿撰写、个性化文献推荐等。随着技术的不断迭代与市场策略的优化，PaperDigest 正逐步实现其愿景，即成为科研工作者在文献理解和论文写作过程中的强大助手，助力学术界迈向更加高效与智能的新时代。

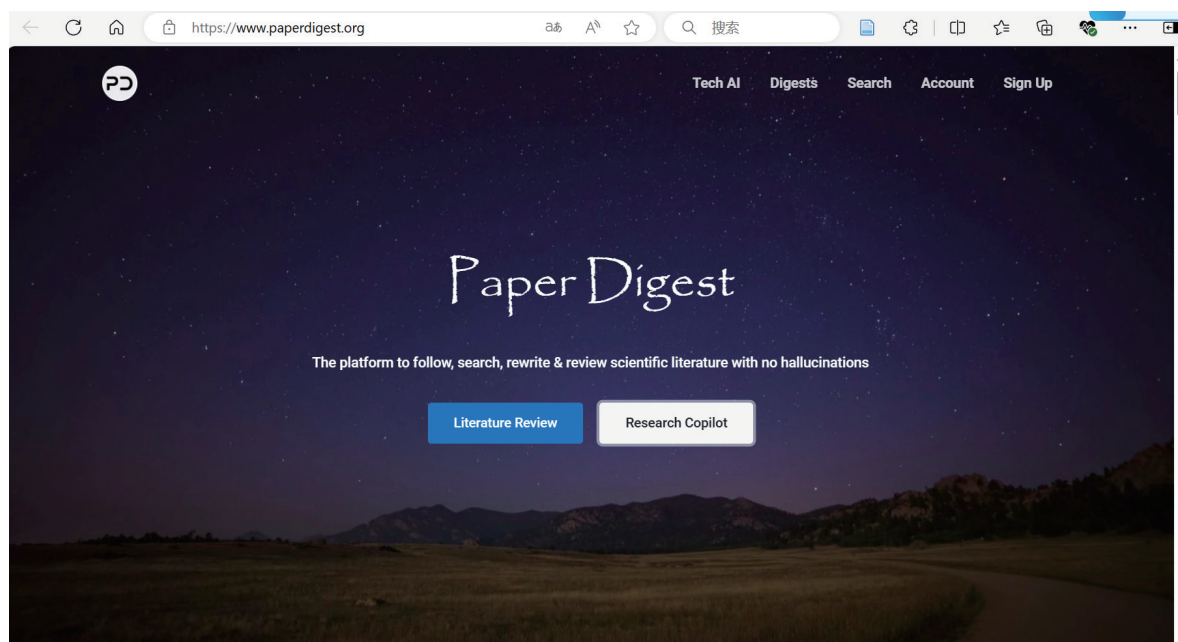


图 5.18: Paper Digest 官网

星火科研助手

星火科研助手，这一由科大讯飞与中国科学院文献情报中心共同研发的创新成果，是科研

领域智能化转型的典范。依托科大讯飞在人工智能与自然语言处理领域的深厚积淀，结合中科院文献情报中心的海量科技文献资源，星火科研助手的诞生旨在为科研工作者提供全方位的智能科研辅助。从星火大模型的孕育之初，科大讯飞便着手打造这款工具，力图运用先进的大模型技术，赋能科研人员在文献调研、论文解读及学术创作中的各个环节。

平台于 2023 年 10 月 24 日首次向社会发布，标志着科研人员可以开始使用这项创新的服务。自发布以来，星火科研助手迅速在中国科学院、三亚崖州湾科技城、北京邮电大学、哈尔滨工业大学等高校和研究机构得到广泛应用和积极反馈。随着时间的推移，星火科研助手经历了多次升级，例如讯飞星火 V3.5 版本的发布，进一步提升了其论文处理能力和科研人员知识获取效率。

具体而言，在文献调研阶段，它能够精准推荐并快速筛选相关文献，提炼核心观点，帮助科研人员及时把握领域动态；在论文解读上，星火科研助手深入解析论文结构，提取关键要素，促进跨文献比较分析，加深对研究议题的理解；至于学术写作，它不仅提供语法和风格的优化建议，甚至能依据研究框架自动生成论文草稿，同时协助文献引用，保证论文质量。

成果调研功能：

- a、综述生成：选择要生成综述的文章，一键生成综述。
- b、学者论文调研：调研指定学者的论文成果。
- c、研究方向论文调研：调研指定研究方向的论文成果。
- d、学者论文总结：对指定学者的论文进行概括总结。
- e、研究方向论文总结：对指定研究方向的论文进行概括总结。
- f、研究方向学者 / 机构推荐：针对指定研究方向推荐学者或机构。

以“量子计算”为例展示成果调研功能的使用。当在搜索框输入量子计算后，星火科研助手将在左侧呈现出相关文献，右侧的对话框是对左侧文献的概要总结。通过选中左侧文献并点击“综述生成”按键，即可快速获得该领域文献的前沿概况。目前，星火科研助手最多支持对 30 篇选中的论文生成综述。用户还可以通过在右侧调研对话框中进行提问来获取研究方向或学者的论文成果情况。例如，可以询问“中科大的潘建伟院士有哪些研究成果？”通过点击末尾的“论文循证”按键，系统可在左侧显示对话框回复内容涉及到的学术论文。

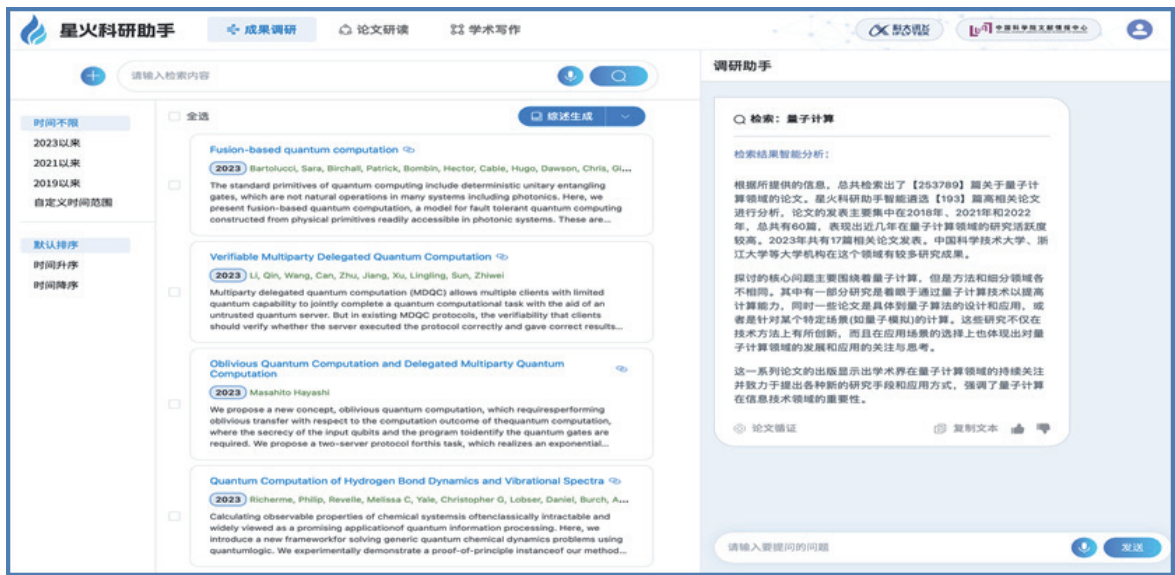


图 5.19: 成果调研使用界面

论文研读功能：

- a、论文知识问答：根据论文中内容进行回答；
- b、论文外问答：开放式问答，需要论文外知识；
- c、片段翻译、总结：对选中片段进行翻译或总结。

以 OpenAI 经典论文 InstructGPT 为例展示论文研读功能。在完成论文上传后，系统会自动用中文给出论文的摘要、方法概述和结论信息，如图 2 所示。在论文研读对话框中，用户可以询问“这篇论文的创新点是什么？”“这篇论文有哪些贡献”等论文内知识，如下图所示。

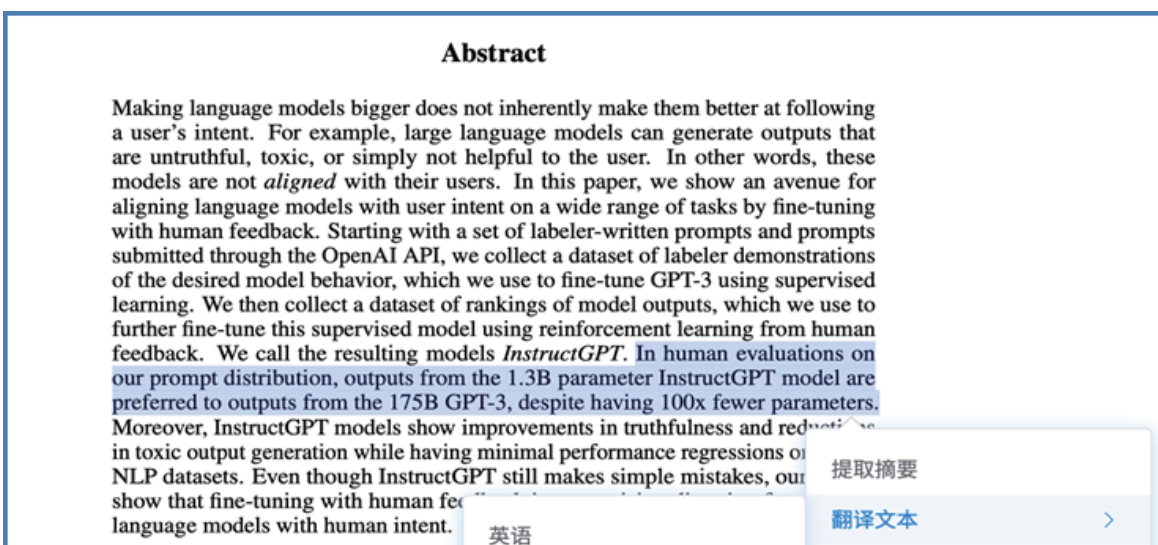


图 5.20: 论文研读使用界面



图 5.21: 论文内知识问答

通过对论文中片段进行选中,系统可以帮助用户获得片段摘要或将片段翻译成中文或英文,并可以在对话框中引用文本进行针对片段的询问。



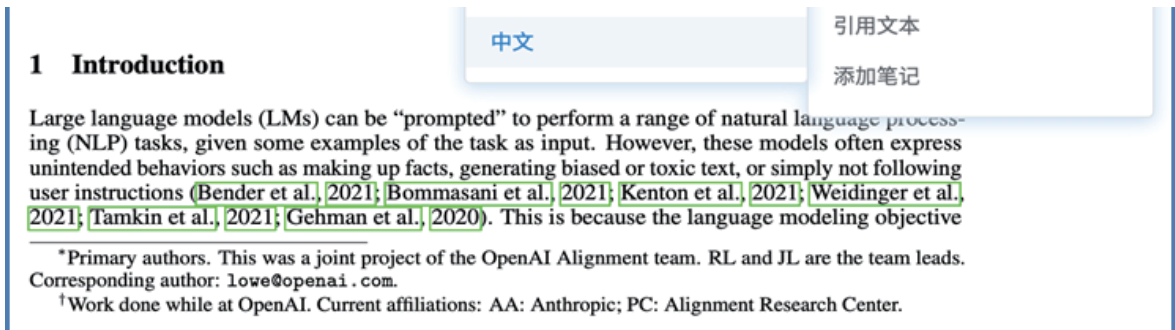


图 5.22: 划线功能展示

除了论文内知识问答，用户也可以在对话框中询问论文外的知识内容，例如“介绍 GPT 模型的发展历程”。



图 5.23: 论文外知识问答

学术写作：星火科研助手中的学术写作功能具有学术翻译能力。以李德毅院士提出的“人工智能基础问题”为例，在点击“学术翻译”后，用户可以通过直接复制文本或上传中文文档，系统会将其翻译为英文，或者用户上传英文文档，系统可以将其翻译成中文。学术翻译功能支持 docx 格式文档的上传，输出结果同样会保存为 docx 文档供用户下载使用。



图 5.24: 文档翻译功能

展望未来，大模型在论文书写辅助领域的应用正揭示出一幅激动人心的图景。技术的不断进步与应用场景的拓展，正推动大模型向着更智能、更个性化的方向迈进。未来的辅助系统将深入理解每位研究者的独特偏好、专业背景和具体需求，提供高度定制化的学术支持，实现前所未有的精准与高效。这一愿景不仅局限于自动化撰写摘要、引言等标准化内容，从而显著减轻研究者的写作负担，节省宝贵时间；更在于它能即时提供反馈与修订建议，扮演一位细致入微的虚拟编辑角色，助力研究者迅速修正文中瑕疵，全面提升论文质量。

与此同时，大模型正逐步消除跨领域、跨语言的壁垒。随着算法精进与数据处理能力的提升，它们将在不同学科间自由穿梭，整合多角度文献资料，为研究者构建起连接各学术领域的桥梁。语言差异不再构成障碍，大模型能够处理多样化的语言文献，扩展研究视野，深化全球学术交流。

然而，在描绘这一美好前景之际，我们亦不可忽视随之而来的挑战——学术诚信与 AI 生成内容的辨识问题。大模型虽有效提升了写作效率，却也带来了确保研究原创性与透明度的新课题。为此，学术界需建立完善机制，识别 AI 生成内容，预防不当使用，以维护学术公正与可信。研究者有责任在利用大模型时保持诚实与审慎，清晰标注 AI 与人工撰写的界限，保障研究透明度。同时，学术期刊、会议组织者及评审专家应更新评价体系，开发精确的检测工具，确保评估过程的公平与准确。

应对这些挑战，学术界与技术开发者应紧密协作，共同制定行业准则，推进先进鉴别技术的研发，并加强对研究者的培训与指导，倡导合理、负责任的大模型使用理念。唯有如此，方能在享受技术红利的同时，守护学术研究的纯净本质与价值，推动知识创新在健康、有序的环境中茁壮成长，共同开启一个既高效又公正的学术研究新纪元。



CHAPTER 6

第 6 章 大模型图片应用

6.1 图片信息提取

当我们谈论大型模型如何处理图片信息时，我们常常想象一个复杂而精密的过程，就像一位专家艺术家在观察世界一样。这些模型不仅能够识别和理解图像中的对象、场景和情感，还能够从中提取深层次的信息。就像调色板上的调色师一样，它们从每一幅图像中提取出色彩、形状、纹理和空间关系，然后用这些信息来“绘制”出图像中的内容。这种能力不仅仅是简单的图像识别，更是对视觉世界的深刻解析，让我们能够更好地理解和互动。

下面从 OCR、识别统计图表并粗略分析、图片转文字三个方面展开详细介绍大模型图像处理信息的技术原理及应用。

6.1.1 OCR

第一个 OCR（Optical Character Recognition，光学字符识别）技术主要用于将图片、扫描文档等中的文字信息转换为可编辑、可搜索的文本格式。其应用十分广泛，包括文档数字化（将纸质文档转换为电子文档，方便存储、搜索和编辑）、身份证识别（从身份证图片中提取姓名、身份证号等关键信息）、票据识别（自动从发票、收据等票据中提取金额、日期等信息）等。OCR 的技术原理是通过图像预处理、文字定位、字符分割、字符识别等步骤，将图像中的文字转换为文本。

6.1.2 统计图表识别

第二个识别统计图表技术是指利用计算机视觉和图像处理技术，自动识别和解析统计图表（如柱状图、折线图、饼图等）中的数据，并进行简单的分析。其应用包括数据分析（自动从图

表中提取数据，减少人工录入时间和计算的工作量)、报告生成(根据图表数据自动生成报告或摘要)，识别统计图表的技术由三部分组成，分别为图像检测、图表解析和数据分析，其中图表检测技术原理是通过图像分割和特征提取，定位图表在图像中的位置。图表解析技术原理是识别图表的类型(如柱状图、折线图等)，并提取图表中的数据。数据分析是对提取的数据进行简单的统计分析，如计算平均值、最大值、最小值等。

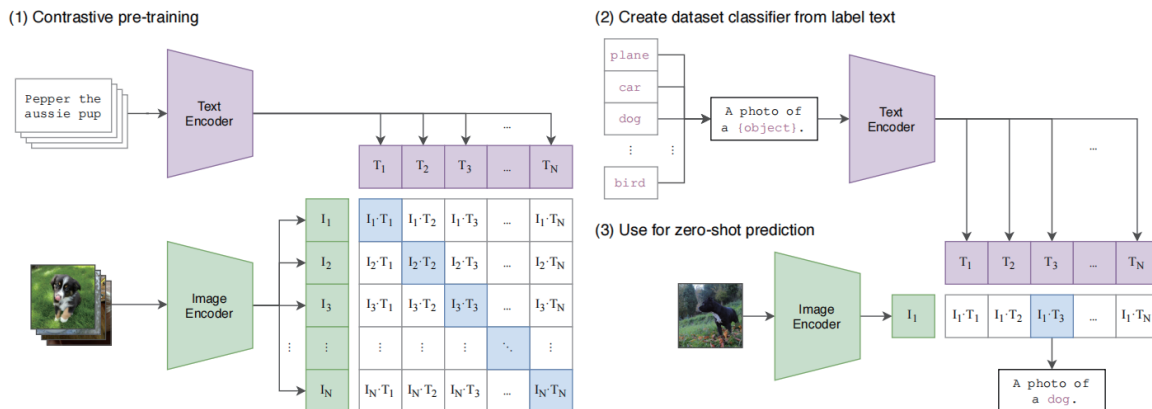


图 6.1: CLIP 模型

6.1.3 图片转文字

第三个是图片转文字技术是将图片中的文字内容转换为可编辑、可搜索的文本格式。其应用包括笔记整理(将手写笔记或打印资料转换为电子文档)、社交媒体内容提取(从社交媒体图片中提取文字信息，如微博、朋友圈等)、书籍扫描(将纸质书籍扫描后转换为电子书籍)。图片转文字的技术原理与 OCR 技术类似，通过图像预处理、文字定位、字符分割、字符识别等步骤，将图像中的文字转换为文本。但图片转文字更注重对复杂背景、字体、排版等的处理能力。

6.2 AI 绘图

AI 绘图是指利用人工智能技术来生成、修改或优化图像的过程。近年来，随着深度学习和计算机视觉技术的发展，AI 绘图在艺术创作、设计、游戏开发、医疗影像分析等领域得到了广泛应用。AI 可以生成独特的艺术作品，帮助艺术家找到新的创作灵感；可以自动生成海报、广告素材、网页设计等，节省设计师的时间和精力；可以辅助医生分析医疗图像，例如 X 光片、MRI 图像等，进行疾病诊断和治疗方案制定；可以修复老旧照片、提高图像分辨率、去除噪声等。

AI 绘图涉及到两个模型：

- **CLIP 模型**：是 OpenAI 开发的一种强大的模型，能够理解和处理图像和文本的多模态信息。CLIP 的设计目的是通过对比学习将视觉和语言信息映射到一个共同的嵌入空间，从而实现图像和文本的联合理解。CLIP 是图像编码器和文本编码器的组合，它可以使得图像编码器和文本编码器提取到相似的特征。两个编码器将图像和文本映射到相同的嵌入空间，通过对比学习，使得描述相同内容的图像和文本在该空间中的距离更近，而描述不同内容的图像和文本距离更远。

- **扩散模型**：是通过一个逐步添加噪声的过程，将数据（如图像）逐渐转换成纯噪声，然后再通过反向过程逐步去噪，恢复到原始数据。这一过程可以分为两个阶段，首先给高清图的图像不断加入随机噪声，直到高清图变成噪声图片，之后用模型预测噪声值，去掉噪声值，生成一个高清的新的内容。训练标签值为加入的噪声。

6.2.1 文生图的操作方法

文生图指的是通过文字描述产生图片，一般用英文单词或短语作为提示词进行描述。文生图平台提供了将文本描述转化为图像的强大工具，这些平台包括在线服务和本地化解决方案。

下面介绍几种主要的文生图平台。

1.DALL-E 2

DALL-E 2 是 OpenAI 在 2022 年 4 月发布的一个 AI 绘图模型。该模型能够根据文本描述生成高质量、细节丰富且具有创意的图像。

• DALL-E 2 的训练过程：

阶段一：CLIP 预训练。使用大规模的图像和文本对数据集进行训练，通过对比损失函数，使得匹配的图像 - 文本对的相似度最大化，而不匹配的相似度最小化。CLIP 学习到如何将图像和文本对齐到相同的嵌入空间。

阶段二：扩散模型训练。利用 CLIP 模型，将文本描述转换为嵌入向量，作为扩散模型的条件输入，从随机噪声开始，通过逐步去噪的过程，生成与文本描述相符的图像。

• DALL-E 2 基本使用步骤：(1) 输入文本描述：用户输入详细的文本描述，描述希望生成的图像内容。(2) 生成图像：模型根据文本描述生成图像，用户可以查看并下载生成的图像。

• 示例：(1) 输入文字描述 “a steampunk airship flying over a city at night”。



图 6.2: DALL-E 生成图像

生成图像为细节丰富的蒸汽朋克飞船，夜晚城市的灯光和飞船的设计细节都非常出色。

(2) 输入文字描述 “an astronaut riding a horse in a futuristic city”。



图 6.3: DALL-E 生成图像

生成图像为结合了科幻和奇幻元素，展示了一个骑马的宇航员和未来城市的场景。

2. Midjourney

Midjourney 是一款 2022 年 3 月面世的 AI 绘图工具。Midjourney 是一种基于 AI 的图像生成工具，能够根据用户提供的文本描述生成高质量的图像。它利用了先进的机器学习模型来理解文本并创建符合描述的视觉内容。官网网址为：<https://www.midjourney.com/>。Midjourney 的使用需要用到 discord 平台，加入 Discord 的 Midjourney 服务器，网址为：<https://discord.com/invite/midjourney>。

- discord 建立私人服务器的步骤：(1) 添加服务器 → 亲自创建 → 仅供我和我的朋友使用 → 设置服务器名称；(2) 点击左上角 Discord 图标，然后点击“寻找或开始新的对话”；(3) 搜索 Midjourney Bot，然后按回车，右键 Midjourney Bot，然后点击个人资料；(4) 点击添加至服务器，选择你创建的服务器，再点击继续，点击右下角的授权；(5) 回到服务器，就能看到 Midjourney 在服务器里面了，接下来就可以用 /imagine 命令生成图片了。

- 使用 Midjourney 生成图像的详细步骤：(1) 访问 Midjourney 平台；(2) 输入文本描述：在主界面上，选择“创建新项目”或类似选项，进入图像生成界面，在文本框中输入你希望生成图像的描述。描述可以是详细的场景、对象或风格，如“一个在日落时分的沙滩上的孤独灯塔”或“未来城市的霓虹夜景”；(3) 设置参数：选择特定的艺术风格或调整参数（如颜色、光线、

细节层次等), 以便生成更符合期望的图像, 有些平台会提供预览功能, 可在生成图像前查看设置效果; (4) 生成图像: 点击“生成”按钮, 提交你的文本描述和设置。系统会开始生成图像, 这个过程可能需要几秒到几分钟, 取决于图像的复杂度和系统负载, 生成完成后, 可以查看生成的图像。Midjourney 通常会提供多个版本的图像; (5) 下载与保存: 从生成的图像中选择满意的版本, 点击下载按钮, 将图像保存到本地设备。还可以在平台上保存项目以便以后访问和修改。

- Midjourney Prompts 基础: Midjourney 提示词基本结构: 一个基本的提示可以是一个简单的单词、短语或表情符号。在 prompt 后面输入对文本的描述。Midjourney 提示词高级技巧: 更高级的提示可以包括一个或多个图像 URL、多个文本短语和一个或多个参数。提示词的内容参考: 主题: 人、动物、人物、地点、物体等; 媒介: 照片、绘画、插图、雕塑、涂鸦等; 环境: 室内、室外、月球、水下、太空等; 灯光: 柔和的、环境光、阴暗的、霓虹灯、演播室灯光等; 颜色: 鲜艳、柔和、明亮、单色、彩色、黑白等; 情绪: 镇静、平静、喧闹、精力充沛等; 构图: 肖像、头像、特写、鸟瞰等。

- 绘图风格参考: MJ 模型的 Stylize (风格化) 参数: Stylize: -s (0-1000)。Stylize 参数越低, 就更符合你提供给 Midjourney 的提示, 但艺术性较差; 高 Stylize 值产生的图像非常具有艺术性, 但与提示的关联性也较少, AI 会有更多的空间自由发挥。Style Low = -s 50, Style Med = -s 100, Style High = -s 250, Style Very High = -s 750。

- Midjourney 进阶使用: (1) remix 模式: 在 /setting 中可以开启 remix 模式, remix 模式开启后再使用 variation; (2) describe 命令: /describe 命令允许用户上传一个图像, 并根据该图像生成四条可能的提示; (3) blend 命令: /blend 命令允许用户上传 2-5 个图像, 然后查看每个图像的概念和美学, 并将它们合并成一张新的图像; (4) 关键词权重: 提示词中添加 :: 表示需要着重考虑提示词的某个部分。默认的权重都是 1, hot::2 dog 表示 hot 权重为 2。权重降低可以使用 hot::-0.5 dog; (5) 图像比例设置: 参数用来更改图像的宽高比, 后面的比例必须使用整数, 默认比例是 1:1; (6) 混沌: -chaos 或 -c, 范围: 0-100, 默认值为 0, 改变结果的多样性, 更高的值产生更多不同寻常和意想不到的结果; (7) 图像权重: -iw, 默认值: 0.25 (0-2), 设置图像提示权重相对于文本权重; (8) 取消: no, 负面提示词, -no 参数相当于把组合 prompt 关键词的权重部分设置为“-0.5”; (9) 品质: -quality 或 -q, 默认值为 1 (0.25-5), 可以以设置 <.25, .5, 或 1>。较高的值使用更多的 GPU 时间, 较低的值使用较少的时间;

- Midjourney 科研作图: Midjourney 在科研领域作图主要是利用它的图生图功能来产生与参考图片类似的图片。目前它还不能够产生可控性很强的图片。

6.2.2 文生图的微调方法

在文生图的生成过程中，经常需要对生成的图像进行调整以满足特定需求。常见的调整方法包括提高图像分辨率和生成图像变体，即对图像进行微调。提高图像分辨率（upscale）用于增强图像的清晰度和细节，而生成图像变体（variation）则用于探索不同的视觉效果和增加图像多样性，以便更好地满足特定应用场景的需求。本节将详细介绍这两种方法。

1. 提高图像分辨率 (U)：提高图像分辨率是为了增强图像的清晰度和细节表现，使图像在放大后仍能保持高质量。这个过程通常被称为“upscale”，在指令中使用“U”表示。具体操作如下：

- U1: 提高第一张图片的分辨率，使其变得高清。
- U2: 提高第二张图片的分辨率，使其变得高清。
- U3: 提高第三张图片的分辨率，使其变得高清。
- U4: 提高第四张图片的分辨率，使其变得高清。

例如，在印刷行业中，设计师需要将客户提供的低分辨率图像放大到适合打印的高分辨率图像，这个过程涉及对图像进行插值放大，即图像插值算法，可以在保留图像细节的同时，减少放大过程中的模糊和失真，这使得最终的印刷品具有更高的质量和更好的视觉效果，这在处理需要高分辨率输出的任务时尤其重要。为了提高图像分辨率，可以选择不同的图像插值算法，如最近邻插值、双线性插值和双三次插值。近年来，基于深度学习的超级分辨率算法（如SRCNN、ESRGAN）也被广泛应用于图像分辨率提升。

- 最近邻插值：这是最简单的插值方法，通过复制最近的像素值来放大图像，适用于对速度要求较高但对图像质量要求较低的场景。这种方法容易导致马赛克效应。

- 双线性插值：这种方法通过线性插值计算新像素值，是一种较为平衡的插值方法，能够在速度和质量之间取得较好的平衡，适用于大多数场景。

- 双三次插值：这种方法通过三次插值计算新像素值，能够提供更高的图像质量，但计算量较大，适用于对图像质量要求较高的场景。

- 基于深度学习的超级分辨率：如SRCNN和ESRGAN，这些方法能够显著提升图像质量，特别是在细节处理上表现优异，但需要较高的计算资源。

2. 图像微调 (V)：图像微调是为了在保持原有图像风格和内容的基础上，生成多个略有不同的变体图像，进行细节上的调整和优化，从而提供更多选择以满足不同需求。这一过程被称为“variation”，在指令中通常使用“V”表示。具体操作如下：

- V1: 对第一张图片进行微调。

- V2: 对第二张图片进行微调。
- V3: 对第三张图片进行微调。
- V4: 对第四张图片进行微调。

微调技术可以包括颜色调整、形状变换、风格迁移等。通过调整图像的亮度、对比度、饱和度等参数，生成不同色调的图像变体。例如，在广告设计中，可以生成不同色调的图像来测试哪种色调更能吸引目标受众。对图像进行旋转、缩放、平移等操作，生成不同形状的图像变体。例如，在产品展示中，可以通过不同角度的图像来全面展示产品特性。使用风格迁移技术，将原始图像的内容与另一张图像的风格相结合，生成风格独特的图像变体。例如，在艺术创作中，可以将现代摄影作品与古典画作风格相结合，创造出独特的艺术作品。

3. 使用 Midjourney 进行微调：在上一节中，介绍过文生图平台 Midjourney，它是一个知名的 AI 生成图像平台，用户可以通过简单的文本描述生成高质量的图像。平台集成了先进的深度学习算法，支持高分辨率图像生成和多种图像微调功能。并且介绍了使用 Midjourney 生成图像的详细步骤。我们继续使用 Midjourney 平台对生成的图片进行微调，以达到我们最终想要的图片效果。在生成图下方我们会发现有一些按钮，右侧蓝色按钮可以进行重新绘制。如果我们点击 U2，不进行分辨率的设置，默认会生成是 1024*1024。

• 示例：



图 6.4: 创建的一张文生图

如果需要进一步放大，点击 Upscale，其中 Upscale (Subtle) 可以理解为，温和的放大，画面内容基本不变；Upscale (Creative) 可以理解为创造性的放大，画面内容可能会有一些小的改变。

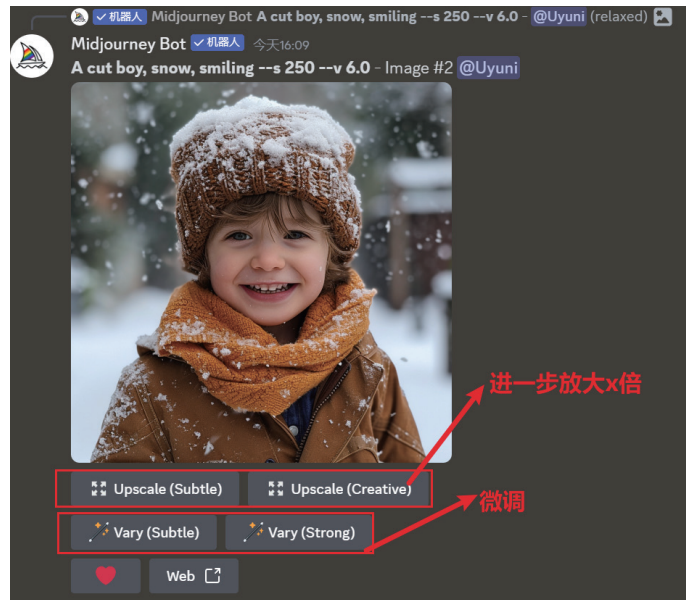


图 6.5: 放大后的第二张图片

V 代表基于已经绘制出来的图片做一些调整。其中 Vary (Subtle) 代表“温和”，也就是微改变一些，Vary (Strong) 代表“强烈”也就是基于已有的图改变较大。

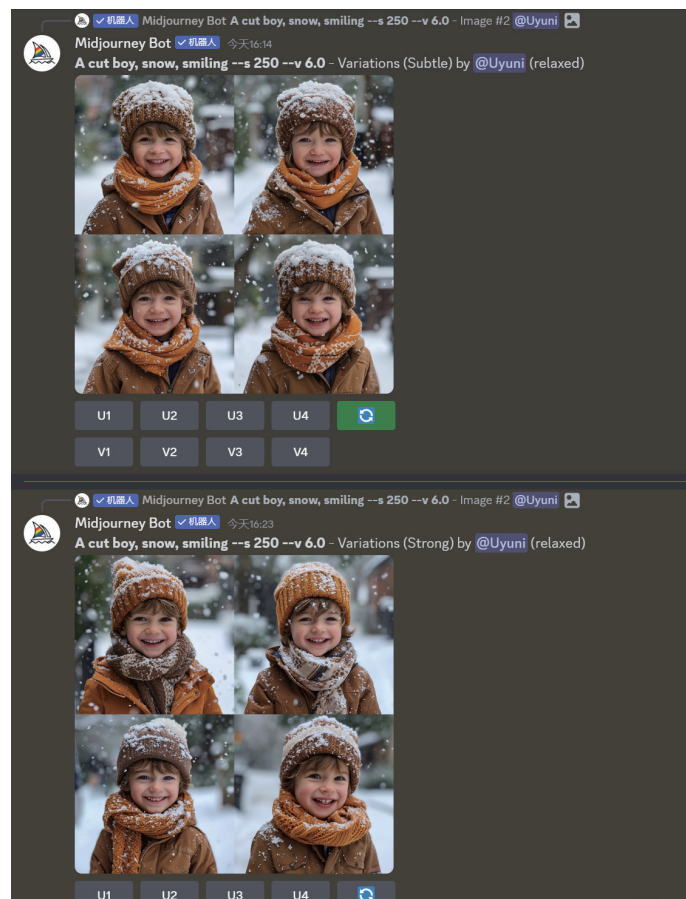


图 6.6: 不同程度的微调对比图

可以对比一下会发现温和的 Vary 帽子围巾什么的基本都没发生变化，而强烈的 Vary 帽子，围巾都发生了变化。

4. 更多应用实例：

- **科学研究与教育**：在科学研究和教育中，图像的高分辨率和准确性至关重要。例如，在生物医学研究中，通过提高显微镜图像的分辨率，可以更清晰地观察细胞结构和组织样本的细节，进而更准确地进行分析和研究。在教育中，高质量的图像可以帮助学生更好地理解 and 掌握复杂的概念和知识。

- **虚拟现实与游戏开发**：虚拟现实（VR）和游戏开发对图像质量有着极高的要求。通过提高图像分辨率和进行微调，可以生成更加逼真的场景和角色，提升用户的沉浸感和体验。例如，在 VR 游戏中，通过对场景图像进行高分辨率处理和风格迁移，可以创建出更具吸引力和代入感的游戏环境。

- **数据增强与机器学习**：在机器学习模型的训练过程中，数据的多样性和质量直接影响模型的性能。通过对训练数据集中的图像进行分辨率提升和微调，可以增加数据的多样性，增强模型的泛化能力和鲁棒性。例如，在图像分类任务中，通过生成不同角度、不同色调的图像变体，可以有效提高模型的分类准确性。

5. 总结：文生图的微调方法是提高生成图像质量和多样性的重要手段。通过合理使用提高分辨率和图像微调技术，可以大大提升图像的视觉效果和应用价值。Midjourney 平台集成了先进的图像处理技术，提供了高效便捷的解决方案，使用户能够轻松实现图像的高清化和多样化。在实际操作中，根据具体需求选择合适的技术和方法，能够事半功倍地达到预期目标。



CHAPTER 7

第 7 章 大模型项目标书

7.1 项目标书的结构

项目标书是一份正式的文档，用于向资助机构、政府部门、学术团体或其他组织申请资金、资源或批准以开展特定的项目或研究工作。它详细描述了项目的背景、目标、研究内容、预期成果、实施计划、预算和时间表等，以说服评审者该项目值得投资和支持。

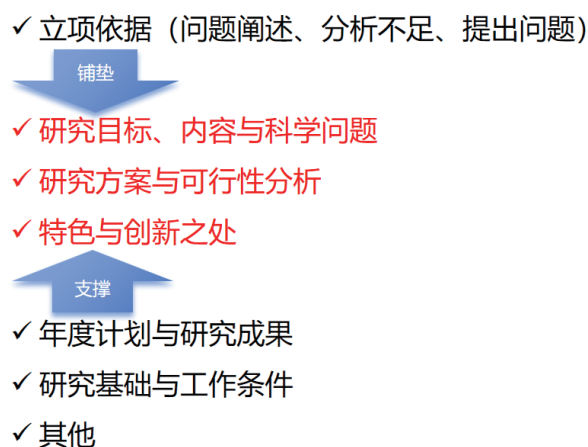


图 7.1: 项目标书的结构

如图所示，一般的项目标书结构包括：立项依据（问题阐述、分析不足、提出问题），研究目标、内容与科学问题，研究方案与可行性分析，特色与创新之处，年度计划与研究成果，研究基础与工作条件等。



通过作图或文档方式明确写作框架

✓ 确定研究内容

前期研究的延续、新颖的思想、内容

✓ 组建团队成员

结构合理、研究基础可覆盖研究内容

图 7.2: 书写思路

在书写项目标书之前，应当提前布局，将个人研究规划与项目申请相结合、相互促进，找专家交流，形成一个思路和规划文档（1-2 页），反复推敲与打磨。

7.2 如何用大模型润色项目标书

7.2.1 项目名称

项目名称是学术思想高度凝练的集中体现，是科学研究水平的最直接反映，在使用大模型提问时，在给出项目总体描述之后，可以这样提出要求：

- 用词准确、简洁
- 能清晰表达最重要和最鲜明的信息
- 能够突出体现项目选题新颖性和创新性
- 长度要适中，在 20-25 字左右
- 切忌题目过大，追时髦

7.2.2 申请代码和关键词

申请代码是申请人在理解代码覆盖范围的基础上，对照项目的科学问题和主要研究内容，选择的合适的学科代码。

关键词则是准确反映科学问题、研究内容、技术方法、研究意义等，依照逻辑关系进行排列的一系列词语。

申请代码和关键词是评审项目分组和专家匹配的重要依据，决定由那些专家评审，对项目的评审有关键影响。

7.2.3 中英文摘要

摘要是整个标书内容的高度浓缩，规定 400 个字，主要回答做什么、为什么做、怎么做及本

研究意义。英文摘要的撰写非常重要，要字斟句酌，简洁明了，避免评审专家在细节上挑毛病。

以下面的中英文摘要为例，在书写摘要时可以使用大语言模型进行润色修改，帮助您更好的完成一篇摘要，可以输入您的研究内容，并给出如下的格式：

• 摘要：

研究背景：……等多重复杂因素与……共演化（影响……），给……带来了巨大的挑战。

研究意义：为应对……，研究如何……，探明……，实现……的条件和机制是……的难点，具有重要的现实意义。

研究目标：本项目旨在分析……，解释……机制，从而解决……

研究内容：本研究研究融入……，分析……，结合……方法，提高模型参数估计、预测和策略评估的精度，解释……机制，甄别……的关键因素。

关键技术和科学问题：提出……，揭示……规律；基于……，发展……方法；提出……方案，统筹……

最后一句：服务于国家……的重大需求。

• Abstract：

Background：The interplay and co-evolution of multiple complex factors such as [.....] and[.....] significantly influence [.....], posing substantial challenges to [.....].

Significance：To address [.....], it is imperative to investigate the mechanisms and conditions underpinning [.....], identify [.....], and establish the pathways to achieving [.....].

This endeavor represents a critical challenge and holds significant practical implications.

Objectives：This study aims to analyze [.....], elucidate the mechanisms of [.....], and provide solutions to [.....].

Content：The study integrates [.....] into the analysis of [.....], leveraging [.....] methodologies to enhance the precision of model parameter estimation, forecasting, and policy evaluation. It seeks to explain the mechanisms of [.....] and identify the key factors driving [.....].

Key Techniques and Scientific Questions：Propose [.....] and reveal the patterns of [.....]; Develop [.....] methodologies based on [.....]; Design [.....] frameworks to comprehensively manage [.....].

Conclusion：This research serves the critical national demands for [.....].

7.2.4 立项依据、国内外研究现状、参考文献

在拟定立项依据的时候，一般采取三段论，设定若干小标题，做到条理清晰、层次分明，

选题的引出和描述应由面到点、逐步深入、逻辑连贯。在使用大模型润色时，可以给出如下的参考格式。

- 研究问题阐述 – 段末总结：问题有挑战性，拟用……，来研究……，服务于……
- 国内外研究现状 – 段末总结：从不同方面进行总结、评述，说出有什么不足；
- 本项目拟做的研究 – 基于上述的不足，总结出关键的科学问题，简述拟研究什么。

国内外研究现状相当于综述、又不同于综述，在书写时应当注意：

- (1) 研究问题是否有意义？是否有科学意义？是否符合国家需求？
- (2) 是否找出了存在的科学问题？综述内容是否分析得当？
- (3) 写作是否清晰、明了？

参考文献是标书的一个重要组成部分，参考文献的引用不仅反映了一个科研人员对本领域学科发展状况的了解程度，也反映了申请人研究层次和水平。除了可以使用本书第五章中提及的论文搜索和阅读方法之外，还需要注意以下几点：

- 参考文献一般 40 篇左右，不宜过少或过多
- 务必要引用本研究领域顶级期刊、最新、最重要的文献
- 可适当引用国内同行的文献，避免低水平重复
- 引用自己以前发表的相关文章，表明自己的研究基础

7.2.5 研究目标、关键科学问题、研究内容

研究目标是项目名称的具体化，是研究对象、研究方法、研究成果和应用的高度概括。应当言简意赅，4-5 行即可，不要重复前面的研究背景，也不要写成研究内容的压缩版。所有研究工作都应围绕研究目标展开，研究内容要与研究目标相互对应，能体现不同标题内容之间的逻辑关联，也使标书结构清晰，便于评审专家阅读。

关键科学问题是研究内容涉及的基础理论和方法的升华，是评审专家关注的重点，一定要仔细推敲，反复琢磨，表达科学、恰当。建议两三个即可，要与自己的申请类型相协调。

专家在评阅研究内容时，主要关注：

- (1) 研究周期内是否能完成；
- (2) 分析研究完成后能否实现研究目标；
- (3) 撰写是否与技术路线重复；
- (4) 是否有递进关系。

在使用大模型润色时，可用以上的要求作为关键词，或者使用第二章中角色扮演的方法将以上要求输入到大模型中令其扮演专家的角色给出指导性意见。

7.2.6 研究方案、可行性分析

研究方案是研究内容实施的各个过程的集合，是对所有研究过程最为详实的表述，是针对研究目标专门设计的，要有鲜明特点、切实可行，要确保能完成研究内容、实现研究目标。不能把研究领域内所了解的基础理论和研究方法统统罗列，也不能泛泛而谈，要强化针对性和可操作性。

可行性分析是说明提出的方案、研究手法是可行的，从理论分析、研究手段/方法分析，对项目组成员搭配以及其运用技术方法的能力分析，可以描述科研团队具有的完成该项目研究的全部条件。

在上述两个部分中，可能存在的问题有：

- (1) 过于简单：在方案中只有方法名称而无具体思想或做法；
- (2) 过于繁杂：大量罗列一些常规的方法；
- (3) 不清楚，不详细，不可行。

在使用大模型润色该部分时，可以让其扮演评审专家的角色，给出如下的关键词帮助评判：

- 看是否是目前最先进、最科学的方法；
- 看是否将其步骤和关键环节体现在技术路线当中；
- 技术路线的主线是否清晰
- 具体步骤是否详细
- 逻辑关系是否清楚

7.2.7 研究特色与创新

研究特色与创新部分是从研究视角、选用方法技术、方案设计、本研究预期结果的科学性以及研究结论的科学意义等角度寻找项目特色与创新性。应反复打磨，描述一定要精炼。建议列出 2-3 个特色和创新之处，不要写太多，否则不是创新或者提过平淡。

可以在向大模型提问时加入如下的关键词：多学科交叉融合特色，建模思想的创新，方法的创新……

7.2.8 年度研究计划、预期研究成果

年度研究计划应按年度列出，与研究内容及其阶段目标对应；应包括拟组织的学术活动、国际合作与交流计划等。

预期研究成果的结构如下，可以将材料和格式交由大语言模型进行整理：

- 理论成果：建立 / 丰富 / 补充 / 填补……
- 技术方法：建立 / 完善……
- 论文：高水平论文
- 人才培养：青年科技骨干、博硕士研究生

7.2.9 总结

在进行项目申请时，除了可以借用大模型的功能外，还需注意以下几点：

- 背景阐述清晰、挑战问题定位明确；
- 研究内容有层次感、递进关系、科学问题 2-3 项；
- 技术路线与思路明确（配图、以创新思路为主）；
- 研究基础好、且可支撑研究内容；
- 写作规范、图文并茂、认真打磨。