

蓝桥杯全国大学生软件和信息技术大赛组委会

第十七届蓝桥杯全国大学生软件和信息技术大赛 人工智能赛（人工智能应用科目） 竞赛规则及说明

一、参赛对象

具有正式全日制学籍并且符合相关科目报名要求的研究生、本科及高职高专学生（以报名时状态为准）。

每位选手配备一名指导教师，同一名指导教师可指导多位选手。全国选拔赛和全国总决赛比赛后指导教师原则上不能更改。

二、组别设置

该科目分为大学组和职业院校组两个组别，每个组别单独评奖。每位选手只能申请参加其中一个组别的竞赛。

研究生、本科生只能报大学组；高职高专学生可自行选报任意组别。

三、竞赛赛制

本届大赛采用校赛、全国选拔赛、全国总决赛三级竞赛体系。

校赛：由参赛院校自行组织并进行评审。

全国选拔赛：由大赛组委会统一组织。比赛时长为 4 小时。

全国总决赛：由大赛组委会统一组织。比赛时长为 4 小时。

详细赛程安排以组委会公布信息为准。

四、竞赛形式

1. 全国选拔赛、全国总决赛均采用封闭、限时方式举办。选手以个人为单位，独立进行作答。

2. 采取线上比赛的形式。

3. 大赛指定竞赛系统，选手机器需访问互联网，以浏览器方式发放赛题。

五、参赛选手机器环境

1.选手机器配置:

(1) 机器硬件要求

最低配置：CPU 处理器：Intel Core i5-5300U 或 AMD Ryzen 5 2600H 以上；内存不小于 4G，硬盘不小于 80G。

推荐配置：CPU 处理器：Intel Core i5-8300H 或 AMD Ryzen 5 3550H 及以上，内存不小于 4G，硬盘不小于 80G。

(2) 摄像头可正常开启，且无遮挡。

(3) 正常连接互联网。

(4) 安装 Chrome 浏览器 v100 以上版本，务必关闭/卸载所有浏览器插件。

(5) 参赛选手可根据不同机型及操作系统自行选择录屏软件（推荐使用 EVCapture 录屏软件、OBS Studio 视频直播录制软件、Mac 自带录屏软件 QuickTime Player 等）。参赛选手需在赛前进行下载安装及调试，比赛前不再提供下载安装及调试时间。如因参赛选手的个人原因未及时下载安装，导致比赛时间耽搁或比赛成绩无效，责任由选手自行承担。

2.在线编程环境:

序号	软件名称及版本号
1	Visual Studio Code 软件版本: v1.101.2 扩展: Chinese、Python、Jupyter
2	python 3.10
3	numpy 2.2.6
4	scikit-learn 1.7.0
5	pandas 2.3.1
6	matplotlib 3.10.3
7	scipy 1.15.3
8	opencv-python 4.12.0.88
9	transformers 4.53.2
10	jieba 0.42.1
11	pillow 11.3.0
12	beautifulsoup4 4.13.4
13	xgboost-cpu 3.0.2
14	spacy 3.8.7
15	plotly 6.2.0
16	onnx 1.18.0
17	onnxruntime 1.22.1
18	Flask 3.1.1

19	torch 2.3.0+cpu
20	torchvision 0.18.0+cpu

说明：表格所列为编程环境中的部分 Python 库。

六、赛题形式

赛题全部为实操任务。选手根据任务要求，通过编写程序代码、完善程序代码的形式完成预期需求。题目均包含完整的题面文档和基础源代码。题面文档中会详细说明题目的背景、需求、目标。选手需认真读题，实现最终目标。

七、赛题范围

本次比赛主要考查机器学习、深度学习、自然语言处理、计算机视觉等人工智能技术应用相关领域职业能力。要求选手根据竞赛题目进行业务需求分析，基于人工智能实训环境，完成数据预处理、模型训练与评估、模型应用部署等工作任务。考查范围详见人工智能应用知识点大纲，链接：dasai.lanqiao.cn/notices/846。

八、答案提交

选手只有在比赛时间内提交的答案内容是可以用来评测的，比赛之后的任何提交均无效。

选手在比赛系统可重复提交自己的答案，以最后一次提交的答案为准并作为评测的依据。

选手须使用比赛指定的方式来提交答案，任何其他方式的提交（如邮件、U 盘）都不作为评测依据。

比赛过程中，赛题分数不会显示给选手，选手应当在没有反馈的情况下自行设计数据调试自己的程序。

选手须仔细阅读并严格遵守赛题指定的答案文件格式或内容。

九、样题

样题详见文档附录。

十、评分

全部题目将使用机器自动评分。评分标准如下：

模块	任务	评分细则	总分值
模块 1	01	完成目标得 6 分。	20
	02	完成目标得 7 分。	
	03	完成目标得 7 分。	

模块 2	04	完成目标得 15 分。	50
	05	完成目标得 20 分。	
	06	完成目标得 15 分。	
模块 3	07	完成目标得 10 分。	30
	08	完成目标得 10 分。	
	09	完成目标得 10 分。	

十一、奖项设置及评选办法

1.全国选拔赛

全国选拔赛各组别设立一、二、三等奖，原则上各奖项的比例为 10%、15%、25%，总获奖比例不超过 50%。获奖比例仅作为参考，组委会专家组将根据赛题难易程度及整体答题情况，制定各奖项获奖最低分数线，未达到获奖最低分数线者不得奖。全国选拔赛一等奖选手获得全国总决赛参赛资格。

2.全国总决赛

总决赛各组别设立一、二、三等奖，原则上各奖项的获奖比例为 10%、25%、40%，总获奖比例不超过 75%。获奖比例仅作为参考，组委会将根据赛题难易程度及整体答题情况，制定各奖项获奖最低分数线，未达到获奖最低分数线者不得奖。

十二、奖项查询

全国选拔赛及全国总决赛评审完成后，大赛组委会将在报名系统开放奖项查询。参赛选手若对奖项有异议，可在 3 个工作日内按照大赛组委会相关要求提出复核申请。

十三、监督反馈

详见《蓝桥杯全国大学生软件和信息技术大赛章程》。

十四、其他注意事项

1. 选手必须符合参赛资格，不得弄虚作假。资格审查中一旦发现问题，则取消其报名资格；竞赛过程中发现问题，则取消竞赛资格；竞赛后发现问题，则取消竞赛奖项，收回获奖证书及奖品等，并在大赛官方网站公示。

2. 参赛选手应严格遵守蓝桥杯大赛比赛管理办法（<https://dasai.lanqiao.cn/notice/s/844/>），服从大赛组委会的指挥和安排，爱护竞赛场地的设备。未尽事宜请参照组委会在大赛官方网站公布的通知、章程、比赛管理办法及相关要求并遵照执行。

3. 赛项采用智能化评审系统自动评分与人工复核双轨校验机制，确保评审效率与精度。选手要特别注意提交答案的形式。必须仔细阅读题目的要求和示例，不得随意添加不需要的内容。

4. 全国选拔赛及全国总决赛前，大赛组委会将在大赛官方网站公布线上比赛手册，请参赛选手及时关注大赛官方网站通知，并按照线上比赛手册要求进行备赛。



附录：

样题

一、背景描述

从党的十八届三中全会提出推动媒体融合发展重大任务，到“十四五”规划建议中明确提出推进媒体深度融合、实施全媒体传播工程、做强新型主流媒体、建强用好县级融媒体中心；从中央办公厅、国务院办公厅印发《关于推动传统媒体和新兴媒体融合发展的指导意见》，到印发《关于加快推进媒体深度融合发展的意见》，媒体融合发展成为国家战略。推进传统媒体和新兴媒体深度融合发展，技术创新是驱动力。人工智能、大数据、云计算、5G、物联网、区块链等核心技术发挥重要作用。

在互联网时代，信息不再稀缺。以新媒体平台为例，随着平台普及和用户规模的不断扩大，大量的数据源源不断地产生和积累，利用人工智能技术来打造智能生产和传播平台已成为一种必要和重要的需求。新媒体平台内容一般呈现出多模态的特点，包括文本、图像、视频等形式。不同模态的数据可以提供丰富的信息表达方式。图像和视频可以传达更直观、生动的视觉信息，而文本则可以提供更加详细和抽象的语义信息。基于人工智能技术，可以充分挖掘数据中的潜在信息，提供更深入的分析和洞察；可以提供更高效的创作者内容生产工具；可以自动检测信息传播的准确性；可以全方位地分析用户的行为、需求、兴趣和情感状态，为企业、组织带来更精准的决策支持和业务价值。

现在，你们是某新媒体平台的研发团队成员。为了使平台提供更加智能化、个性化的用户体验，你们将结合自然语言处理、计算机视觉等领域的人工智能技术，以 Python 作为基础开发语言，利用传统机器学习和深度学习算法进行新媒体数据的相关处理、分析、应用。你们作为该团队的核心成员，请按照下面任务完成本次工作。

二、成果物提交

人工智能应用开发赛项参赛选手按照各模块的赛题要求提交对应的成果物。

答案提交目录为 `/home/project/answer`。参赛选手务必在该目录下存放每题的成果文件夹，否则将导致提交失败，影响得分。

选手需在 `/home/project/answer` 目录下创建 01、02、...、09 文件夹，然后将各题的成果文件放在对应文件夹内，最后提交。

本赛项基于选手提交的竞赛成果物，进行评分。所有提交内容的文件命名务必与要求的文件命名保持一致，否则将被记为零分。

需提交的成果物如下：

模块	赛题序号	成果物
模块 1	01	songs_processed.csv 文件
	02	task.py 文件
	03	task.py 文件
模块 2	04	songs_testout.csv 文件
	05	houses_testout.csv 文件
	06	pred_test.txt 文件
模块 3	07	task.py 文件
	08	task.py 文件、text_classifier.onnx 文件
	09	task.py 文件

1. 缺陷数据预处理

【介绍】

在当今数字音乐时代，大量的歌曲被发布，但只有少数能够进入热门排行榜。对于音乐产业而言，能够准确预测一首歌曲的热度具有重要的商业意义。本任务提供了一个关于音乐的数据集，它包含了大量的歌曲信息以及与之相关的各种特征，基于这些特征，我们能够使用机器学习算法来预测歌曲的热度。但是，这个数据集存在缺陷，需要你们研发小组对这个数据集做一些预处理。songs_origin.csv，是本任务提供的数据集。数据集中包含了丰富的音乐元数据和特征，如声学特性、舞蹈性、能量等。

【目标】

请按以下要求实现对数据集的预处理。

- ① 处理数据集中的缺失值，对于数据集中的缺失值，以其所在列的均值进行填充。
- ② 处理数据集中的异常值，对于 acousticness_yr 列的值大于 1 或小于 0 的行进行删除。
- ③ 处理数据集中的重复行，对于数据集中出现多行的相同数据，只保留一行，删除其余重复行。

请在 01 文件夹下 task.py 文件中编写代码，针对数据集 songs_origin.csv 完成以上数据处理，并将处理后的结果保存在模块一文件夹下，命名为 songs_processed.csv。正确实现以上对数据集的处理，即完成目标。

2. 图像数据预处理

【介绍】

在医学影像分析中，准确的图像预处理对后续的诊断至关重要。本任务提供了一批医学 CT 扫描图像，你的团队需要使用 `OpenCV` 库对这些图像进行必要的预处理，为后续的病变检测做准备。`MedicalData.zip` 是本任务提供的数据集。解压后，你将看到一个 `patient_info.json` 文件和一个 `scans` 文件夹。其中，`patient_info.json` 文件包含了患者信息，`scans` 文件夹包含了多个子文件夹，每个子文件夹对应一位患者的 CT 扫描序列。

【目标】

请按要求编写以下函数代码。

`ct_preprocessor` 函数

✓ 函数功能

使用 `OpenCV` 对输入的 CT 扫描图像进行预处理，包括图像去噪、对比度增强和归一化。

✓ 参数

`scan_path`: 字符串，指定 CT 扫描图像文件的路径。

`output_size`: 元组，指定最终输出图像的目标尺寸，默认为 `(512, 512)`。

✓ 返回值

`image_original`: `numpy.ndarray`，原始 CT 扫描图像。

`image_processed`: `numpy.ndarray`，预处理后的 CT 扫描图像，尺寸为 `output_size`。

`noise_level`: `float`，去噪前图像的噪声水平估计值。

请在 02 文件夹下 `task.py` 文件中 `#TODO` 处补充代码，确保实现以下目标。

(1) 图像读取：使用 `cv2.imread()` 函数读取 CT 扫描图像。

(2) 图像去噪：使用 `cv2.fastNlMeansDenoising()` 函数对图像进行去噪处理，并使用 `cv2.meanStdDev()` 函数估计原始图像的噪声水平。

(3) 对比度增强：使用 `cv2.equalizeHist()` 函数进行直方图均衡化来增强图像对比度。

(4) 归一化：使用 `cv2.normalize()` 函数将图像像素值归一化到 `[0, 1]` 范围内。

(5) 尺寸调整：使用 `cv2.resize()` 函数将图像调整为指定的 `output_size` 大小。

3. 数据增强

【介绍】

数据增强是提升新媒体平台中自然语言处理（NLP）模型性能的关键技术。在这些平台上，用户生成的文本数据多样且丰富，但也往往包含有限的语义表达方式。通过数据增强，我们可以创造出各种训练样本的变体，这不仅扩大了训练数据集，还增强了模型对新颖表达

方式的理解能力，从而提升了模型在实际应用中的泛化能力。同义词替换就是一种有效的数据增强方法，它通过使用同义词典替换训练数据中的词汇，以生成语义相似但表达不同的训练样本。在本任务中，考生需要在 PyTorch 的 Dataset 类中实现这样一个数据增强方法，借助提供的同义词典，实现同义词替换功能。这种方法尤其适用于新媒体平台，可以帮助模型更好地适应并理解用户的多样化语言表达，进而提高新媒体平台上各类 NLP 应用的性能。

data.csv 是训练数据，包含 160 条样本，其中 text 列表示文本内容，text_id 列表示文本对应的 id。loc.txt 和 per.txt 是本任务提供的地名和称谓增强字典，共包含了 273 个地名和 66 个家庭称谓。

【目标】

请按要求编写以下函数代码。

augment 函数

✓ 函数功能

对输入的文本进行数据增强。

从地名/称谓增强词典中，找出所有出现在输入文本中的地名/称谓。

对于每个找到的地名/称谓，从增强词典中随机选择一个新的地名/称谓作为替代，并将文本中的原地名/称谓替换为这个新地名/称谓。

替换信息记录在 aug_info['locs'] 和 aug_info['pers'] 列表中，包括原始地名/称谓和替换后的地名/称谓。

✓ 参数

text 输入文本

✓ 返回值

增强后的文本和增强信息。

aug_info['locs'] 和 aug_info['pers'] 为嵌套字典列表，其中列表中每个元素及其说明见下表。

字段名	释义	数值类型
original	原始实体	字符串
replacement	替换后的实体	字符串

请在 03 文件夹下 task.py 文件中 #TODO 处补充函数代码，并运行，确保能够实现以下目标：

- (1) 正确地返回增强后的文本和增强信息。
- (2) 除了对增强词典中的替换，其他文本保持不变。

4. 回归预测模型训练

【介绍】

音乐对于许多人来说是一种情感表达和情绪释放的方式。音乐流媒体平台为用户提供了广泛的音乐内容和社交互动功能。通过分析歌曲的特征和历史数据，可以估计哪些歌曲可能受用户欢迎、应该最大程度地提高哪些歌曲的曝光度。本任务提供了一个关于音乐的数据集，它包含了大量的歌曲信息以及与之相关的各种特征。目前，我们已经对这个数据集做了一些预处理和特征选择。请你们研发小组基于这些特征，使用机器学习算法来实现自动预测歌曲的热度。`songs_train.csv` 是本任务提供的训练集。其中，`popularity` 列是目标变量，其余是有关音乐的特征，如声学特性、舞蹈性、能量等。`songs_test.csv` 是本任务提供的测试集，与训练集同分布。其中，`popularity` 列是目标变量（该列数值为空），其余是有关音乐的特征，如声学特性、舞蹈性、能量等。

【目标】

请在 04 文件夹下 `task.py` 文件中编写代码，按以下要求实现对歌曲热度的自动预测。

- (1) 选择合适的回归算法，构建一个回归模型。
- (2) 使用训练好的回归模型预测测试集 `songs_test.csv` 的目标变量。
- (3) 将输出结果保存在模块二文件夹下，命名为 **`songs_testout.csv`**。注意：结果文件中保留测试集 `songs_test.csv` 的原始数据，仅填充 `popularity` 列即可。
- (4) `songs_testout.csv` 结果得分不低于 0.8，视为实现目标。

提示：判题过程中的得分计算，以回归模型的 R 平方（决定系数）为准。

5. XGBoost 房价预测模型优化

【介绍】

在房地产市场中，准确预测房屋价格对于买家、卖家和投资者都至关重要。通过分析房屋的各种特征和历史交易数据，可以估算出房屋的市场价值。本任务提供了一个关于房地产的数据集，包含了大量的房屋信息以及相关特征。我们已经对这个数据集进行了初步的预处理和特征工程。请你们研发小组这些特征，使用 XGBoost 算法来实现对房屋价格的精确预测，并重点关注模型的泛化能力。

`houses_train.csv` 是本任务提供的训练集。其中，`price` 列是目标变量，其余是房屋的特征，如面积、卧室数量、地址位置等。`houses_test.csv` 是本任务提供的测试集，与训练集同分布。其中，`price` 列是目标变量（该列数值为空），其余是房屋特征。

【目标】

请在 05 文件夹下 `task.py` 文件中编写代码，按以下要求实现对房屋价格的精确预测。

- （1）使用 XGBoost 算法构建回归模型，重点关注防止过拟合的策略。
- （2）实现梯度提升过程，合理设置学习率和迭代次数。
- （3）使用训练好的 XGBoost 模型预测测试集 `houses_test.csv` 的目标变量。
- （4）将输出结果保存在模块二文件夹下，命名为 `houses_testout.csv`。注意：结果文件中保留测试集 `houses_test.csv` 的原始数据，仅填充 `price` 列即可。
- （5）`houses_testout.csv` 结果的均方根误差（RMSE）不高于 50000，且 R 平方不低于 0.85，视为实现目标。

提示：

判题过程中的得分计算，以回归模型的均方根误差（RMSE）和 R 平方（决定系数）为准。

重点考虑 XGBoost 的正则化参数（如 `max_depth`, `min_child_weight`, `gamma` 等）来防止过拟合。

合理使用早停（early stopping）策略来优化模型训练过程。

可以尝试使用交叉验证来调整超参数，提高模型的泛化能力。

6. 虚假信息识别模型训练

【介绍】

新媒体作为信息传播的重要渠道，需确保传播的信息可信可靠。因此，平台需要采取假新闻过滤机制限制虚假信息的传播，提高内容质量。在本任务中，你将基于处理好的数据集，设计并构建一个针对新闻文本的真假分类模型。`news_train.txt` 是本任务提供的训练集文本。文件中每行是一个样本，即经过分词处理后的句子。`label_newstrain.txt` 是本任务提供的训练集标签。文件中每行是一个 0/1 数值标签，与 `news_train.txt` 文件逐行一一对应。0 表示文本真实，1 则表示虚假。`news_test.txt` 是本任务提供的测试集文本。文件中每行是一个样本，即经过分词处理后的句子。测试集的 F1 值不低于 0.9，即视为实现目标 1。

【目标】

请在 06 文件夹下 `task.py` 文件中编写代码，并按以下要求实现对新闻文本的真假进行自动预测。

- (1) 选择合适的分类算法，构建一个分类模型。
- (2) 使用训练好的分类模型预测测试集 `news_test.txt` 的类别标签。
- (3) 将输出结果保存在模块二文件夹下，命名为 `pred_test.txt`。注意：结果文件中只保存 0/1 标签，每行一个数值，第一行的数值对应 `news_test.txt` 中第一行文本的真假类别，以此类推。
- (4) `pred_test.txt` 结果准确率不低于 0.9，视为实现目标。

提示：判题过程中的准确率计算示例：真实标签为 `[0, 1, 0, 1, 1]`，预测标签为 `[0, 1, 1, 1, 0]`，准确率为 0.6，表示 60% 的样本被正确预测。

7. 多模型融合与优化

【介绍】

你的研发团队已经开发了多个用于预测股票价格的机器学习模型。为了提高预测的准确性和稳定性，你需要对这些模型进行融合和优化。在本任务中，你将使用多个训练好的模型，实现模型融合，并通过交叉验证来优化融合权重，以获得最佳的预测结果。

【目标】

`model_lstm.h5`、`model_xgboost.pkl`、`model_random_forest.pkl` 是本任务提供的三个不同类型的预训练模型。

请按要求编写以下函数代码。

`optimize_ensemble` 函数

✓ 函数功能

加载提供的 `model_lstm.h5`、`model_xgboost.pkl`、`model_random_forest.pkl` 模型。

使用交叉验证方法优化三个模型的融合权重。

使用优化后的权重对测试数据进行预测。

✓ 参数

`x_train`: `numpy.ndarray`，训练数据特征。

`y_train`: `numpy.ndarray`，训练数据标签。

`x_test`: `numpy.ndarray`，测试数据特征。

✓ 返回值

`best_weights`: `list`，最优的模型融合权重。

`y_pred`: `numpy.ndarray`, 使用最优权重融合后的测试集预测结果。

`cv_score`: `float`, 交叉验证过程中得到的最佳评分。

请在 07 文件夹下 `task.py` 文件中 `#TODO` 处补充函数代码, 确保实现以下目标:

- (1) 正确加载三个不同类型的模型 (LSTM、XGBoost、Random Forest)。
- (2) 实现一个权重优化函数, 使用网格搜索或贝叶斯优化等方法, 通过交叉验证找到最优的模型融合权重。
- (3) 使用优化后的权重对测试数据进行预测。
- (4) 返回最优权重、测试集预测结果和交叉验证的最佳评分。

8. 模型转换

【介绍】

在前面的任务中, 我们搭建并训练完成了文本分析模型。在实际应用中, 还需要将训练后的模型转换为其他格式, 以支持不同的推理引擎。ONNX (Open Neural Network Exchange) 提供了一个开放的源模型格式, 支持多种深度学习框架之间的模型转换。本任务要求你们研发小组将提供的基于 PyTorch 的文本分类模型转换为 ONNX 格式, 并实现一个使用 ONNX 模型进行推理的简单应用。`model.pt` 是本任务提供的 PyTorch 模型权重文件。

【目标】

请按要求编写以下函数代码。

`convert` 函数

✓ 函数功能

将本任务提供的 `pt` 模型文件转换为 ONNX 格式。

ONNX 模型文件保存在模块三文件夹下, 命名为 `text_classifier.onnx`。

`inference` 函数

✓ 函数功能

读取 ONNX 模型文件。

使用 ONNX 模型进行推理。

返回推理结果。

✓ 参数

`model_path`, 字符串类型, ONNX 模型文件的绝对路径。

`input`, 整数列表类型, 为待预测样本, 示例如: `[101, 304, 993, 1008, 102]`。

✓ 返回值

result, 浮点数列表类型, 为 ONNX 文件推理结果, 示例如: `[[0.53419, 0.44313]]`。

请在 08 文件夹下 `task.py` 文件中 `#TODO` 处补充函数代码, 并执行 `main()` 函数, 确保能够实现以下目标:

- (1) 正确转换模型文件。
- (2) 正确输出推理结果。

9. 模型部署

【介绍】

为了充分挖掘新媒体平台上积累的用户文本数据中的命名实体信息并为其提供应用服务, 我们需要将该模型部署到线上环境。一般来说, 在自然语言处理应用的开发中, 我们会使用 Web 框架 (如 Flask) 部署模型, 并为外部提供 API 接口以支持在线预测。在本任务中, 你们研发小组将接手一个部分完成的 Flask 项目。该项目已经包括了命名实体识别 (NER) 模型的载入、运行以及接收 `tokenize` 后的数据的代码。考生的核心任务是将模型的整数向量输出转换为一个结构化的实体标注列表, 并通过 Flask API 返回这个列表。此步骤不仅实现了自动化的实体识别, 更使得新媒体平台可以实时、高效地获取文本中的实体信息, 为后续的数据分析和运营决策提供有力支撑。`ner.pt` 是本任务提供的 PyTorch 模型权重文件。

【目标】

请按要求编写以下函数代码。

`process` 函数

✓ 函数功能

将本任务提供的请求数据使用提供的模型进行推理。

将推理结果转换为结构化的实体标注列表。

对于不合理的序列, 如单字实体 `['O', 'B-LOC', 'O']` 和以 `I-` 起始的实体 `['O', 'I-LOC', 'I-LOC']`, 将其作为非实体序列处理。

实体标注列表为一个嵌套字典列表, 其中列表中每个元素及其说明见下表。

字段名	释义	数值类型
start	实体的起始位置	数值
end	文本的结束位置	数值
label	实体类型	字符串

请在 09 文件夹下 `task.py` 文件中 `#TODO` 处补充函数代码，并运行代码块，确保能够实现以下目标：

- （1）正确地使用模型进行推理。
- （2）返回正确的结构化结果。