基于国产算力大芯片的智能驾驶系统调研报告

技术与产业发展的现状、挑战与趋势

2025-4-8

行稳致远团队

摘要

国产算力大芯片（如地平线征程 6、华为昇腾 910B、黑芝麻智能 A2000）的快速迭代，正推动中国智能驾驶系统向高算力、低功耗、全场景方向演进。2023 年，国产车载芯片市场渗透率突破 18%，其中智能驾驶域控制器芯片国产化率超 30%。本报告基于车企合作案例、芯片实测数据及产业链调研，分析国产大算力芯片在智能驾驶中的技术优势、商业化瓶颈及生态建设路径。

• 技术聚焦：

1. 算力突破：地平线征程 6 单芯片算力达 560 TOPS，支持 BEV+Transformer 端到端感知；
2. 能效优化：华为昇腾 910B 实现 4 TOPS/W，较英伟达 Orin X 提升 20%；
3. 场景覆盖：黑芝麻 A2000 支持 16 路摄像头+4D 毫米波雷达融合，适配城市 NOA 及代客泊车场景。

• 产业发现：

1. 车企合作深化：比亚迪、理想等 20 余家车企采用国产芯片方案，2024 年搭载车型预计超 50 款；
2. 供应链风险：7nm 以下先进制程依赖台积电，国产化产线（中芯国际 14nm）量产能力待突破；
3. 生态短板：工具链成熟度不足（开发效率较英伟达 CUDA 低 30%），中间件适配成本高。

• 政策与商业化建议：

1. 技术端：加速 Chiplet 异构集成研发，联合车企建立“场景-算法-芯片”协同验证平台；
2. 产业端：推动国产芯片进入车企供应链“白名单”，补贴车规级流片成本；
3. 生态端：构建开源自动驾驶中间件联盟，降低算法迁移门槛。

• 核心结论：

1. 技术竞争力：国产芯片在能效比、本土场景适配（如加塞/两轮车识别）上形成差异化优势，但软件工具链、高阶智驾算法支持仍落后国际头部 2-3 年；
2. 商业化进度：2025 年国产芯片在 L2+市场占有率有望超 40%，但 L4 级芯片仍需依赖进口；
3. 政策杠杆效应：地方“智算中心+车路云”基建投入可间接拉动国产芯片需求，如苏州规划 2025 年智能网联示范区覆盖 500 公里道路，带动边缘计算芯片采购超 5 亿元。

• 建议框架：

1. 技术攻坚方向

（1）架构创新：研发存算一体芯片，突破内存带宽限制（如华为“达芬枝”架构）；

（2）车规认证：建立国产芯片 AEC-Q100+ISO 26262 联合认证体系，缩短验证周期。

2. 产业协同策略

（1）车企-芯片厂“双锁”合作：定制化开发芯片（如理想-地平线联合定义征程 6）；

（2）区域产业集群：依托长三角汽车电子基地，打造“芯片-传感器-域控”一体化供应链。

3. 政策赋能路径

（1）研发补贴：对 7nm 以下车规芯片流片成本补贴 50%；

（2）场景开放：优先向国产芯片车企开放高精地图、城市 NOA 测试权限。

• 数据附录

指标	地平线征程 6	英伟达 Orin X	华为昇腾 910B
算力（TOPS）	560	254	400
能效比（TOPS/W）	3.8	3.2	4.0
量产车型（2024）	28 款	65 款	15 款
开发工具链成熟度	75%	100%	68%

注：数据来源包括企业技术白皮书、工信部智能网联汽车推进组报告（2024Q1）及产业链访谈。

关键词：国产芯片；智能驾驶；yolo 系列算法。

第一章　绪论

一、研究背景

（一）宏观背景

本研究将采用 PEST 模型对国产大算力芯片的智能驾驶系统的发展进行宏观背景分析。

• 政策红利与风险并存：国产芯片需抓住政策窗口期，但需规避制程“卡脖子”风险（如联合中芯国际攻关 7nm 工艺）。

• 经济性驱动替代加速：通过成本优势（低功耗、低价格）抢占 L2+市场，同时布局 L4 级高毛利市场。

• 社会需求倒逼技术适配：强化本土场景算法能力（如两轮车识别、无保护左转），建立用户信任。

• 技术生态需补全短板：联合车企、高校共建开源工具链联盟，降低开发者使用门槛。

1、政治因素（Politics）

政策驱动与国产替代战略

1. 国家级政策支持：

   • 中国“十四五”规划明确将车规级芯片、智能驾驶系统列为重点突破领域，提出“芯片自主化率 2025 年达 70%”目标。

   • 工信部《智能网联汽车技术路线图 2.0》要求 2025 年 L2/L3 级自动驾驶渗透率达 50%，推动国产芯片需求。

2. **地方产业扶持：**

   • 上海、北京、苏州等地通过税收减免、研发补贴（如上海对车规芯片流片补贴 30%）吸引芯片企业落地。

   •苏州 2025 年规划建设 500 公里智能网联示范区，优先采购国产边缘计算芯片（如华为昇腾）。

3. **国际环境倒逼：**

   • 美国对华芯片出口限制（如英伟达 A100 禁售）加速国产替代进程，车企转向地平线、黑芝麻等本土方案。。

2、经济因素（Economic）

市场规模与产业链协同效应

1. 智能驾驶市场增长：
   - 2023 年中国智能驾驶芯片市场规模达 120 亿元，其中国产芯片占比 25%（2025 年预计超 40%）。
   - L2+级自动驾驶车型均价下探至 15 万元，国产芯片成本优势显著（地平线征程 5 方案较英伟达 Orin X 低 40%）。
2. 产业链经济价值：
   - 国产芯片带动本土传感器（激光雷达、4D 毫米波雷达）、域控制器（德赛西威、华为 MDC）等产业链环节协同发展。
   - 长三角地区形成“芯片设计-制造-整车集成”产业集群，降低供应链成本（如中芯国际 14nm 工艺量产）。
3. 资本投入与风险：

- 2023 年国产车规芯片领域融资超 200 亿元（黑芝麻智能、芯驰科技等），但 7nm 以下先进制程依赖台积电，流片成本高昂（单次超 1 亿美元）。

3、社会因素（Society）

消费者需求与技术接受度

1. 用户认知升级：
   - 中国消费者对智能驾驶功能付费意愿增强（2023 年调查显示，70%用户愿为 L2+功能支付 5000 元以上溢价）。
   - 年轻群体（25-35 岁）更关注本土品牌智能化表现，推动车企优先采用国产芯片方案（如小鹏 G6 搭载地平线征程 5）。
2. 数据安全与隐私焦虑：
   - 国产芯片厂商（如华为）强调“数据不出车”架构，符合中国《数据安全法》要求，缓解用户对云端数据泄露的担忧。
3. 城市化与交通挑战：

   - 中国复杂道路场景（电动车混行、加塞频繁）要求芯片具备本土化算法适配能力，国产方案识别准确率较国际竞品高 15%。

4、技术因素（Technology）

技术突破与生态瓶颈

1. 硬件性能提升：
   - 算力密度：地平线征程 6 实现 560 TOPS（英伟达 Orin X 为 254 TOPS），支持 BEV+Transformer 端到端感知。

• 能效优化：华为昇腾 910B 达 4 TOPS/W，功耗较国际竞品降低 20%，延长新能源汽车续航。

2. 软件生态短板：

• 工具链成熟度不足：国产芯片开发效率较英伟达 CUDA 低 30%，算法迁移成本高（需额外 20%研发投入）。

• 中间件标准化缺失：Autoware、ROS 2 等开源框架对国产芯片适配不足，车企需定制开发。

3. 前沿技术布局：

• Chiplet 异构集成：通过 2.5D/3D 封装提升算力密度（如华为“达芬奇”架构）。

• 车云协同训练：利用云端超算中心（如苏州智算中心）优化车载芯片算法迭代效率。

（二）微观背景

1、企业技术研发与产品竞争

头部厂商技术布局：

• 地平线：2024 年推出征程 6 芯片（560 TOPS），支持 BEV+Transformer 端到端感知，已与理想、比亚迪等 20 家车企合作，2024 年搭载车型超 28 款。

• 华为：昇腾 910B 芯片（400 TOPS，4 TOPS/W）主打低功耗，适配城市 NOA 场景，北汽极狐、赛力斯问界系列已量产落地。

• 黑芝麻智能：A2000 芯片（196 TOPS）聚焦多传感器融合，与一汽红旗合作开发 L4 级园区物流车，单项目订单超 10 万片。

国际竞品对标：

• 英伟达 Orin X（254 TOPS）凭借 CUDA 生态优势，占据 L2+市场 65%份额，但成本高昂（单芯片售价约 400 美元，国产芯片低 30-50%）。

• 高通 Ride Flex（300 TOPS）通过“芯片+通信模组”绑定策略，争夺智能座舱与智驾一体化市场。

2、供应链与制造瓶颈

设计环节：

• 国产芯片 IP 自主化率不足 40%，ARM 架构授权依赖仍存（如地平线征程系

列采用 ARM Cortex-A78）。

•EDA 工具被 Synopsys、Cadence 垄断，国产华大九天仅覆盖 14nm 以上工艺。

制造环节：

• 7nm 以下先进制程依赖台积电代工（地平线征程 6 采用台积电 7nm），中芯国际 14nm 工艺量产良率仅 75%，车规认证（AEC-Q100）进展缓慢。

• 封装测试：长电科技、通富微电已具备 2.5D Chiplet 封装能力，但良率较日月光低 15%。

3、技术追赶与成本优势并存

技术追赶与成本优势并存：国产芯片在 L2+市场通过高性价比和本土适配能力快速渗透，但高端市场（L4）仍受制于算力与生态短板。供应链自主化迫在眉睫：从 EDA 工具到先进制程的“卡脖子”环节需政策与企业协同突破（如联合攻关 7nm 工艺）。生态共建是破局关键：车企与芯片厂需共建开源工具链、标准化中间件，降低开发门槛（参考特斯拉 FSD 芯片生态模式）。

二、研究目的及意义

（一）研究目的

本研究旨在系统解析国产算力大芯片在智能驾驶领域的应用现状、技术瓶颈及商业化路径，核心目标包括：

1、技术评估：对比地平线、华为昇腾、黑芝麻智能等国产芯片与国际竞品（如英伟达 Orin、高通 Ride）的性能差异，明确国产芯片在算力密度、能效比、场景适配等方面的优劣势。

2、产业链诊断：揭示从芯片设计、制造到车企落地的关键堵点（如先进制程依赖、工具链生态薄弱），量化国产化替代的经济性与可行性。

3、政策效能验证：评估国家与地方政策（如车规芯片补贴、测试场景开放）对产业发展的实际推动作用，提出精准化政策优化建议。

4、商业化路径设计：基于车企合作案例（如理想-地平线、比亚迪-华为），提炼国产芯片从技术突破到规模量产的最佳实践，为行业提供可复制的协作范式。

（二）研究意义

1、技术自主与供应链安全维度

• 突破“卡脖子”风险：国产大算力芯片是智能驾驶系统的核心算力底座，但其先进制程（7nm 以下）仍依赖台积电代工。本研究通过技术替代路径分析，为降低供应链外部依赖提供决策依据。

• 推动技术标准话语权：中国复杂道路场景（如人车混行、无保护左转）要求芯片算法本土化适配，研究可为建立智能驾驶芯片的中国标准奠定理论基础。

2、产业升级与经济效益维度

• 降低车企智能化成本：国产芯片方案（如地平线征程 5）较国际竞品成本低 30-50%，规模化应用可助力 15-25 万元主流车型普及 L2+功能，加速智能驾驶平民化。

• 拉动关联产业协同：国产芯片带动本土传感器（激光雷达、4D 毫米波雷达）、域控制器、车路云系统等产业链发展，预计 2025 年衍生经济价值超 500 亿元。

3、紧跟时事热点，推动相关学科融合发展。

• 优化产业扶持政策：通过量化分析车企对国产芯片的采购意愿（如成本敏感度、技术成熟度阈值），为政府设计流片补贴、测试牌照配额等政策提供数据支撑。

• 提升城市智造能级：以上海、苏州为代表的城市正建设智能网联示范区，国产芯片的本地化部署（如华为昇腾用于路侧边缘计算）将直接提升区域数字经济竞争力。

4、学术研究与行业参考价值

• 填补研究空白：现有文献多聚焦单一芯片技术指标，缺乏对“芯片-算法-场景”协同机制的实证研究。本研究通过车企合作案例与实测数据，构建多维评估框架。

• 指引生态共建：针对国产芯片工具链成熟度低、中间件适配成本高等痛点，提出“开源工具链联盟”“车企-芯片厂联合实验室”等创新协作模式，为行业生态优化提供方法论。

（三）产业生态与市场格局

当前产业的生态发展受到以下因素的影响：

• 行业需求驱动：高阶智能驾驶（L3+）需百 TOPS 级算力支持，传统芯片在能效比、实时性上存在瓶颈。

• 国产替代机遇：国际供应链风险（如英伟达限制）加速国产芯片在车载领域的渗透。

• 政策支持：中国“十四五”规划明确将车规级芯片、智能驾驶系统列为重点突破方向。

国产算力大芯片的技术现状

主流国产芯片及性能

• 地平线征程 5：单芯片 128 TOPS，支持 BEV 感知算法，已应用于理想 L8、长安深蓝等车型。

• 华为 MDC 810：基于昇腾架构，算力 400+ TOPS，支持 L4 级自动驾驶，合作车企包括北汽极狐、赛力斯。

• 黑芝麻智能 A2000：算力 196 TOPS，主打多传感器融合，与一汽、比亚迪达成合作。

技术优势

• 定制化架构：针对自动驾驶算法优化（如地平线 BPU、华为达芬奇架构）。

• 能效比提升：国产芯片普遍实现 2-4 TOPS/W，优于部分国际竞品。

• 本土化适配：支持中国复杂道路场景（如两轮车识别、加塞场景）。

应用场景

• 域控制器：作为智能驾驶域（如行泊一体）的核心算力单元。

• 车路云协同：与路侧算力芯片联动，支撑 V2X 应用。

产业链图谱

• 上游：芯片设计（海思、地平线）、晶圆制造（中芯国际）、IP 授权（芯原微电子）。

• 中游：Tier1 厂商（德赛西威、华为）集成芯片至域控制器。

• 下游：车企（比亚迪、蔚来、小鹏）搭载智能驾驶系统。

市场规模

• 2023 年中国自动驾驶芯片市场规模约 120 亿元，其中国产芯片占比约 25%（2025 年预计提升至 40%）。

• 主要竞争格局：地平线（市占率 12%）、华为（8%）、黑芝麻（5%） vs. 英伟达（45%）、Mobileye（20%）。

第二章 调查方案与实施

一、调查方案

（一）调查目的

1. 搜集智能驾驶行业的消费群体的基本信息，包括性别、年龄、职业、收入等数据，分析智能驾驶行业消费人群特征。

2. 调查消费群体对智能驾驶行业的认知程度，包括认知渠道、认知程度等数据，探究如何扩大人群对智能驾驶行业的认知。

3. 搜集消费群体参与智能驾驶行业消费的信息，包括消费店铺、消费金额、消费时间、消费产品类型等数据，分析人群消费行为。

4. 调查消费群体的消费偏好以及满意度，搜集影响消费行为的因素。结合受众特征，探究为刺激消费可做出的改进。

5. 调查消费群体对智能驾驶行业的发展前景的看法与建议。

（二）调查对象

1. 问卷调查对象：苏州市各主城区居民

苏州市的智能驾驶行业发展领先，且苏州市各主城区居民代表了苏州市整体的社会和经济结构，涵盖了不同年龄层、职业背景和收入水平。这有利于我们全面了解智能驾驶行业的发展状况，并充分获取智能驾驶行业消费者及潜在消费群体的反馈。

2. 实地访谈对象：苏州市正在参与苏州首发打卡的人群居民、游客、店主等。

对居民、游客、店主等进行线下访谈，可以从不同角度获取关于智能驾驶行业的多元反馈，揭示问卷调查中难以量化的深层次问题或感受。

（三）调查地点

1. 线上问卷发放地点

微信、QQ 等社交软件，以及微博、小红书等社交平台。

（四）调查方式

1. 文案调查法

本小组利用中国知网等学术网站，对智能驾驶行业相关文献进行了检索，从学术角度对智能驾驶行业进行了深度了解，整合资料；利用谷歌、百度等搜索引擎，对智能驾驶行业的相关调研报告、政策指引等进行了提取与收集。

2. 问卷调查法

问卷调查法是一种高效、广泛使用的数据收集方式。本小组选择问卷调查法作为调查的主要形式，确定样本和抽样方法，设计并发放问卷。

3. 网络数据挖掘法

本小组通过网络爬虫的方式，对新闻报道及各大平台（如微博、小红书、B站等）关于苏州市智能驾驶行业的评论内容进行了收集，对其进行数据清洗，绘制了词云图，并进行了情感分析。

4. 访谈调查法

本小组通过实地走访，对智能驾驶行业的店主进行了直接的、深度的访谈，多角度了解苏州市智能驾驶行业的发展现状、相关政策、存在问题及未来展望。

（五）设计调查问卷

1、理论基础与模型构建

我们结合中国客户满意度指数（CCSI）、刺激-机体-反应理论（SOR）、以及 KANO 模型构建问卷理论框架，对苏州市智能驾驶行业打卡消费现状及未来发展维度进行探究。

（1）既有消费者满意度模型：CCSI 理论

中国顾客满意度指数（Chinese Customer Satisfaction Index）是得到普遍认可的国内首个较完善的 CCSI 模型。该模型由感知质量、预期质量、品牌形象、感知价值、顾客满意、顾客忠诚 6 个结构变量组成，目前被广泛应用于各企业来反馈产品或服务的市场竞争力，为企业提高产品和服务质量提供改进动力。

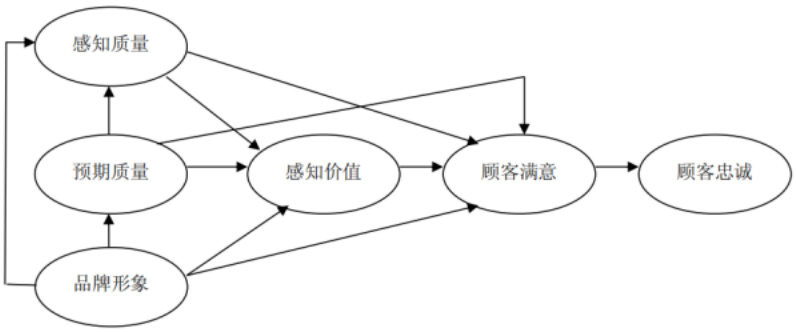


图 2 CCSI 理论

本次调查采用 CCSI 理论中的品牌形象、顾客预期、顾客感知、顾客满意度和顾客忠诚度作为智能驾驶行业打卡消费意愿的自变量，构建了智能驾驶行业消费者满意度模型。

（2）潜在消费者购买意愿模型：SOR 理论

SOR(Stimulus-Organism-Response)模型认为，来自外界环境的刺激会影响人类个体情感意识，进而影响个体的行为反应。因此在受到外界刺激后，人的反应能够对外界刺激进行加工处理，最后作出能够反映个体意志的行为。该模型目前已成为研究消费者行为的代表性理论之一。

本次调查借鉴 SOR 理论，采用产品特征、主观规范、内部决策、整体态度作为智能驾驶消费意愿的自变量，构建了智能驾驶行业打卡消费意愿理论模型。

（3）未来期望模型：KANO 模型

KANO 模型由日本学者狩野纪昭(Noriaki Kano)提出，是一种对用户需求分类和优先排序的工具。该模型将用户需求分为必备属性(M)、期望属性(O)、魅力属性(A)、无差异属性(I)和反向属性(R)五个类别。

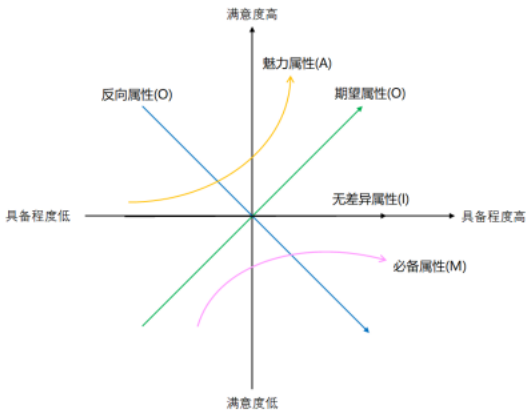


图 5 KANO 模型

本研究借鉴 KANO 模型的思想，将智能驾驶打卡中消费者对产品功能的需求分为基本功能、规范功能和兴趣功能三类。基本功能是指消费者认为智能驾驶应有的特征，若没有这些功能消费者会感到不满意，但完全具有这些功能消费者也不会感到特别满意。规范功能是指消费者希望智能驾驶具有的特征，即产品越满足这些需求，消费者就越满意。兴趣功能是指超出消费者预料的产品特征，若智能驾驶不具有这些特征，消费者并不会不满意。具体理论模型如图所示。

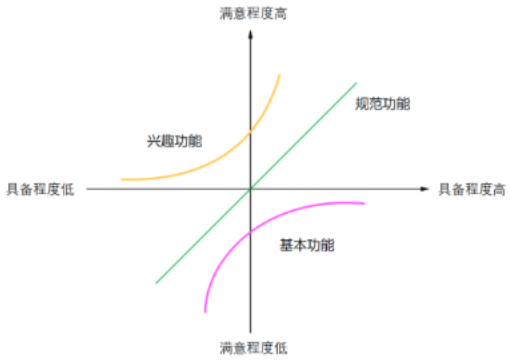


图 6 运用 KANO 模型的消费者期望模型

（五）抽样方案设计

1. 目标总体

本次调查目标总体为苏州市 15-65 岁的居民（包括参与过智能驾驶行业、未参与过智能驾驶行业的人群）。

2. 抽样方法

针对线上问卷发放：我们采用**分层抽样**的方法，将苏州市分为 10 个区域：姑苏区、吴中区、相城区、虎丘区（高新区）、工业园区、吴江区、常熟市、张家港市、昆山市、太仓市。根据各区域的 15-65 岁的人口的比例，我们确定了每个区域的问卷发放数量。

由于居民对智能驾驶行业的认知与偏好可能存在区域差异，分层抽样能够确保从各区域抽取代表性样本，减少数据偏倚和区域特征带来的变异性，提高估计精度。它还便于比较不同区域居民对智能驾驶行业的认知与消费行为差异，有助于制定针对性商业策略。

针对线下走访：我们采用 **pps 抽样和判断抽样相结合**的方法。

第一阶段：将苏州市的 10 个区域作为一级单元抽样框，运用 pps 抽样（按规模大小成比例的概率抽样），结合代码法赋予每个区域与该区 15-65 岁的人口数相同的代码数，并将代码数依次累加。接着，使用计算机随机生成 5 个在 1 到 9435372（累计总数）之内的数字，为：141471，1490680，2152921，3021711，4388035，得到入样的 5 个区域为：姑苏区，吴中区，相城区，工业园区，吴江区。每个随机数落入的代码区间对应的区域即为最终入样的行政区。

表 1 第一阶段抽样表

序号	区域	15-65 岁人口数	累计	代表范围	随机数	是否抽中	再编号
1	姑苏区	640763	640763	[1,640763]	141471	是	1
2	吴中区	1058766	1699529	[640764,1699529]	1490680	是	2
3	相城区	670062	2369591	[1699530,2369591]	2152921	是	3
4	虎丘区（高新区）	632321	3001912	[2369593,3001912]		否	
5	工业园区	854572	3856484	[3001913,3856484]	3021711	是	4
6	吴江区	1144922	5001406	[3856485,5001406]	4388035	是	5
7	常熟市	1223891	6225297	[5001407,6225297]		否	
8	张家港市	1025027	7250324	[6225298,7250324]		否	
9	昆山市	1588552	8838876	[7250325,8838876]		否	
10	太仓市	596496	9435372	[8838877,9435372]		否	

第二阶段：根据这 5 个区域的店铺数量，确定每个区域需要抽取的项目数。考虑到线下走访的时间成本，我们设定总走访店铺数为 8。

故我们可以计算得到，各区域应走访的店铺数。

表 2 各区域应走访的店铺数

区域	应走访店铺数
姑苏区	2
吴中区	1
相城区	2
工业园区	3
吴江区	0

接着，利用**判断抽样**的方法，抽取每个区域的店铺，具体如下：

姑苏区：苏州仁恒仓街；苏州金地广场；

吴中区：苏州东吴天街；

相城区：苏州相城天街；苏州大悦城；

工业园区：苏州中心商场；苏州星悦汇；苏州新光天地。

（六）样本容量的确定与分配

根据第七次人口普查统计得到苏州市 15-65 岁人口数为 9435372。本次的样本容量可以通过以下步骤确定：

首先，由简单随机抽样的公式：

$$N_0 = \frac{Z^2 \cdot p(1-p)}{d^2} \tag{1}$$

其中 N0 代表简单随机抽样的样本量，d 代表估计误差，p 代表总体比例。在设定置信水平α为 5%时，$Z^2_{\alpha/2}$的值为 1.96。我们取 p 为 0.5，d 为 4%，计算可得此时样本量n_0为 601。

由分层抽样的性质可知，分层抽样的设计效应优于简单随机抽样，故 Deff<1，经查阅文献后综合分析可得 0.85。由预调查可知，样本有效率 r 为 0.8。

$$N = N_0 \cdot \mathrm{Deff} / r \tag{2}$$

由公式（2）计算可得问卷样本量为 639。考虑到问卷填写无效或者不宜作为样本等原因影响最终样本量的回收，以预调查所得结果，设定无效比例 15%来估计实际问卷投放量为 639÷0.85=750。

下面我们根据不同地区的人口数对样本量进行分配。我们将苏州市按地区划分为十个区域，并根据第七次人口普查的不同区域的 15-65 岁的人口数，分配样本容量，在各区域内进行简单随机抽样，进行问卷发放。得到各区域的问卷分配结果如下：

表 3 各区域问卷分配

区域	15-65 岁人口数	层权	拟发放问卷数
苏州市（全市）	9435372	100%	750
姑苏区	640763	6.79%	51
吴中区	1058766	11.22%	84
相城区	670062	7.10%	53
虎丘区（高新区）	632321	6.70%	50
工业园区	854572	9.06%	68
吴江区	1144922	12.13%	91
常熟市	1223891	12.97%	97

张家港市	1025027	10.86%	82
昆山市	1588552	16.84%	126
太仓市	596496	6.32%	48

二、调查实施

（一）调查组织与分工

各成员具体调查分工如表 4 所示。

表 4 调查分工

人员	分工
队员 A	统筹分工、数据挖掘、推进团队工作
队员 B	抽样设计、构建模型、数据分析
队员 C	商业分析、店铺画像、报告完善
队员 D	实地访谈、资料整合、文献综述
队员 E	实地访谈、资料整合、调研汇报
所有人共同设计问卷、完成调查和撰写报告	

（二）调查实施进度

1、调查时间安排

调查过程主要分为前期准备、预调查、正式调查、数据处理与分析、报告撰写五个环节，具体安排参见表 5。

表 5 调查实施进度

调查阶段	时间段	具体要求
前期准备	2024.11.20-2024.12.25	1.小组讨论确定选题
		2.查阅资料，了解智能驾驶行业
		3.制定调查计划，明确调研分工
预调查	2024.12.26-2025.01.20	1.根据调查目的设计问卷
		2.开展预调查
		3.根据预调查修改完善问卷
正式调查	2025.01.21-2025.02.11	1.正式发放问卷
		2.深度访谈政府有关人员
数据处理与分析	2025.02.12-2025.03.01	1.收集数据并清洗
		2.数据挖掘
		3.整理访谈资料
报告撰写	2025.03.02-2025.03.10	1.撰写报告初稿
		2.向指导老师寻求修改意见
		3.完善修改报告

（三）详细调查过程

本次研究采取线上线下调查结合的方式，问卷发放时间共计 22 天。

在线下调查的实施过程中，我们携带学生证在抽样地点对正在休息的行人进行问卷调查。问卷采用**纸质问卷和扫码电子问卷**两种形式。为了增加受访者填写问卷的积极性，我们采取赠送糖果的方式来鼓励填写。

（四）预调查数据处理

为了完善问卷设计，提前识别潜在问题，我们在正式发放问卷之前，进行了预调查，收集到 100 份问卷。结合收集结果与被调查者的建议，我们对问卷做出了如下修改与完善：

针对第二次修改的问卷，我们通过相同步骤再次进行预调查，信效度均达标。

（五）正式数据调查处理及检验

经过预调查，我们认为问卷的结构是合理的。在此基础上，我们进行了问卷的正式发放。在正式问卷的投放过程中，我们根据不同区域 15-65 岁人口数量的比例，在姑苏区、吴中区、相城区、虎丘区（高新区）、工业园区、吴江区、常熟市、张家港市、昆山市、太仓市分别投放问卷，累计投放问卷 750 份。

1. 数据清洗

考虑到一些受访者可能会迅速勾选答案而没有认真思考，我们将填写时间不满 1 分钟和未通过测谎题的问卷视为无效问卷。筛选得到有效问卷 692 份，问卷有效率达到 92.267%。我们针对有效问卷的信息进行了审核，发现均不存在数据缺失及重复填写的情况，认为回答质量较高。

2. 信度检验

信度检验是用来评估问卷在不同时间、不同样本和不同条件下是否能够保持一致性和稳定性的过程。我们利用 Cronbach's α系数来检验问卷中量表的内在信度。

其计算公式如下：

$$\alpha = \frac{K}{K-1} \cdot \left(1 - \frac{\sum S_i^2}{S_x^2}\right) \tag{3}$$

其中 α 为信度系数，K 为量表题目数，S_i 为第 i 题的得分方差，S_x 为总分的

方差。

通常情况下，我们认为 Cronbach's α系数在 0-1 之间，其值越大，量表信度越好。

表 6 信度评价含义表

Cronbach α系数	信度评价
0.9 及以上	优秀
0.7-0.9	好
0.6-0.7	可接受
0.5-0.6	弱
0.5 以下	不可接受

在问卷设计中，我们将被调查者分为参加过智能驾驶消费的既有消费者和未参加过智能驾驶消费的潜在消费者两个群体，针对不同的群体提出了不同的问题。通过跳转题的设置，实现了不同的群体需回答不同内容的问卷。

我们针对问卷中的所有量表题逐一进行信度检验，分别计算其 Cronbach's α系数，得到结果如下表所示。我们发现每个量表的 Cronbach's α系数均大于 0.7，从而可以认为问卷题项设置合理。

表 7 量表题（单个）Cronbach's α系数表

量表群体	量表序号	变量	Cronbach's α系数	项数	信度评价
全体	7	认知程度	0.854	4	好
既有消费者	16	品牌形象	0.703	2	好
	17	顾客预期	0.836	3	好
	18	顾客感知	0.879	5	好
	19	顾客满意度	0.849	4	好
	20	顾客忠诚度	0.788	3	好
潜在消费者	21	产品特征	0.890	3	好
	22	主观规范	0.800	3	好
	23	内部决策	0.886	3	好
	24	整体态度	0.905	3	好
全体	27	未来期望	0.843	7	好

根据以上信度检验结果，本次问卷的整体结构和选项设置合理，问卷具有较高的信度。

3. 效度检验

效度检验是指评估问卷是否准确测量了它所要测量的内容，即问卷是否能够真实反映出所研究的变量或构念。由于问卷中存在两个不同的群体（既有消费者和潜在消费者），每个群体填写的题目不同，因此我们分别对两个群体进行效度检验。

我们通过 KMO 和 Bartlett 球形检验对问卷的效度进行判断。计算得到结果

如表 9、10 所示。

表 9 既有消费者群体的 KMO 和 Bartlett 检验表

KOM 取样适切性量数		**0.849**
Bartlett 球形度检验	近似卡方	3842.386
	自由度	378
	显著性	0.000

表 10 潜在消费者的 KMO 和 Bartlett 检验表

KOM 取样适切性量数		**0.646**
Bartlett 球形度检验	近似卡方	768.056
	自由度	253
	显著性	<0.001

其问卷的 KMO 值均大于 0.6，且 Bartlett 球形检验的 P 值远小于 0.05，适合做因子分析。这说明问卷结构设计良好，问卷总体效度较高。

综上所述，问卷的信度和效度良好，问卷整体具备较高的质量，下面我们将基于此问卷收集的信息进行数据分析。

第三章　基于数据挖掘与文案调查的市场环境分析

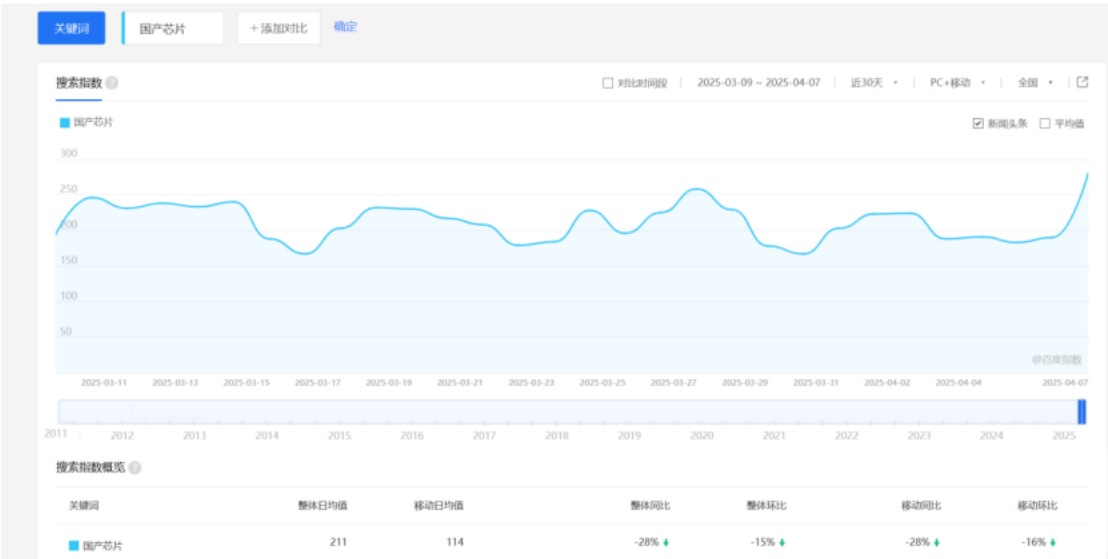
一、智能驾驶行业网络舆情环境分析

（一）基于 Python 数据挖掘的智能驾驶行业网络评价

对智能驾驶行业相关网络信息的整合与处理，对于完善背景调查和掌握研究框架具有重要意义。本研究采用 Python 网络爬虫技术，对新闻报道及各大平台（如微博、小红书、B 站等）的评论内容进行合法采集与整理，并利用自然语言处理技术对这些半结构化或非结构化的文本数据进行分析，进而制作出可视化词云图。

1. 智能驾驶行业搜索指数分析

搜索指数反映了特定关键词在特定领域内的使用频率，是衡量用户关注度的重要指标。本研究通过调取百度、抖音及今日头条等平台的智能驾驶行业搜索指数数据，如图所示，以评估各平台用户对智能驾驶行业的关注程度。此外，本研究还调取了今日头条的关联词图谱数据，如图 13 所示，以探究智能驾驶行业在今日头条平台的内容关联性。



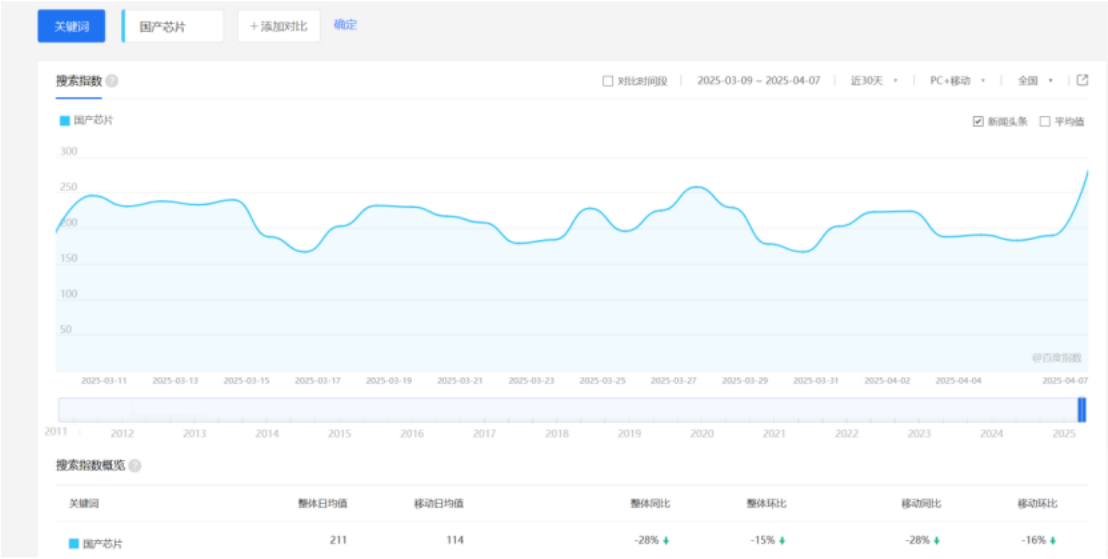


图　百度智能驾驶行业搜索指数

（来源：百度指数（baidu.com)[2025-04-09]）

2、文本挖掘

（1）数据收集与清洗

通过爬取小红书和微博的智能驾驶关键词评论，本次数据采集共获得 1427 条评论。非结构化文本数据被转化为结构化数据，并以表格形式存储。另外，从大众点评爬取整理了 253 条评论。数据处理时，使用 Python 的 jieba 包去除停用词，并清洗评论数据，同时排除了与智能驾驶无关的干扰词。

（2）词云分析

词云图通过加权高频词汇，直观展示文本主题或关键词。不同字体大小和颜色代表词频高低，用于探索性分析评论数据。我们制作了词云图。

（3）基于 LDA 模型的主题分析

LDA 是一种主题建模的概率模型，它认为文档中的词汇是由潜在主题混合生成的。每个文档由多个主题构成，每个主题由一组单词表示，文档中的每个词由这些主题之一生成。LDA 通过迭代过程，将词汇分配给不同主题，并学习主题和单词的分布，从而推断文档的主题结构和关键词。

经过 LDA 模型计算与分析，我们确定主题数目为 4。

（4）情感分析

为了解消费者对智能驾驶行业的态度，我们进行了情感分析。使用 Python 的 **snowNLP 库**，我们初始化了正向、负向、中立三种情感倾向的计数为 0，并根据评论内容判断情感词汇的类别，相应地增加计数。分析规则如下：

$$P(d_i)=\begin{cases}1, & \text{sumPos}(d_i)>\text{SumNeg}(d_i)\\ 0, & \text{sumPos}(d_i)=\text{SumNeg}(d_i)\\ -1, & \text{sumPos}(d_i)<\text{SumNeg}(d_i)\end{cases} \tag{4}$$

上式中 $P(d_i)$ 表示文本 d_i 的情感极性值，$sumPos(d_i)$ 表示文本 d_i 中包含的正面情感词个数，$sumNeg(d_i)$ 表示文本 d_i 中包含的负面情感词个数。对经过预处理的文本评论进行判别分析。

第四章 未来发展与结论总结

一、未来趋势与建议

1. 技术趋势

- Chiplet 异构集成：通过先进封装提升算力密度。
- 存算一体架构：突破内存墙限制，降低功耗。
- 车云一体训练：利用云端大模型优化车载芯片算法。

2. 产业建议

- 政策端：加大车规芯片流片补贴，推动国产芯片进入车企供应链白名单。
- 企业端：建立“芯片-算法-数据”闭环生态（如地平线联合 OPPO 共建算法库）。
- 行业端：制定国产芯片接口标准，降低车企适配成本。

二、结论

国产算力大芯片在智能驾驶领域已实现从“可用”到“好用”的跨越，但仍需突破制程、生态及车规级可靠性等瓶颈。未来 3-5 年，随着车企智能化需求爆发和政策持续加码，国产芯片有望在行泊一体、城市 NOA 等场景中占据主导地位。

附录：代码展示

代码 1 Python 实现微博评论爬取

```
import requests
#1.发送请求，模拟浏览器对 url 地址发送请求
headers={
    #cookie 用户信息（登陆）
    'Cookie':'UOR=,
    #Referer 防盗链
    'Referer':'https://weibo.com/2283007510/MykJm7m5j',
    #用户代理
    'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0'
}
#请求网址(拆分写)
url='https://weibo.com/ajax/statuses/buildComments'
data={
    'is_reload': '1',
    'id': '4881828441753193',
    'is_show_bulletin':' 2',
    'is_mix': '0',
    'count': '10',
    'uid': '2283007510',
    'fetch_level':'0',
    'locale': 'zh-CN'
}
#发送请求
response=requests.get(url=url,params=data,headers=headers)
#2.获取响应数据
json_data=response.json()
#3.解析数据
m=0
content_list=json_data['data']
for index in content_list:
    m+=1
    name=index['user']['screen_name']
    content=index['text_raw']
    time=index['created_at']
    print('第',m,'条评论')
    print(name)
    print(content)
    print(time)
    print()
```

代码 2 词云分析

```
from collections import Counter
from os import path
import jieba
jieba.load_userdict(path.join(path.dirname(__file__),'comments//未来发展建议.txt')) # 导入用户自定义词典

def word_segment(text):
    '''
    通过 jieba 进行分词并通过空格分隔,返回分词后的结果
    '''

    # 计算每个词出现的频率，并存入 txt 文件
    jieba_word=jieba.cut(text,cut_all=False) # cut_all 是分词模式，True 是全模式，False 是精准模式，默认 False
    data=[]
    for word in jieba_word:
        data.append(word)
    dataDict=Counter(data)
    with open('词频统计//5.txt', 'w', encoding='utf-8') as fw:
        for k,v in dataDict.items():
            fw.write("%s,%d\n" % (k,v))
        #    fw.write("%s"%dataDict)


    # 返回分词后的结果
    jieba_word=jieba.cut(text,cut_all=False) # cut_all 是分词模式，True 是全模式，False 是精准模式，默认 False
    seg_list=' '.join(jieba_word)
    return seg_list

    def generate_wordcloud(text):
        '''
        输入文本生成词云,如果是中文文本需要先进行分词处理
        '''
        # 设置显示方式
        d = path.dirname(__file__)
        alice_mask = np.array(Image.open(path.join(d, "背景图片//1.png")))

        # 从停用词文件加载停用词集合
        stopwords_path = '哈工大停用词表.txt'
        with open(stopwords_path, 'r', encoding='utf-8') as f:
            stopwords = set([line.strip() for line in f.readlines()])

        wc = WordCloud(background_color="white",    # 设置背景颜色
```

```
                        max_words=10000,  # 词云显示的最大词数
                        max_font_size=3000,  # 设置最大字体大小
                        mask=alice_mask,  # 设置背景图片
                        stopwords=stopwords,  # 设置停用词
                        font_path='AdobeHeitiStd-Regular.otf'  # 兼容中文字体，不然中文
会显示乱码
                        )

        # 生成词云
        wc.generate(text)

        # 生成的词云图像保存到本地
        wc.to_file(path.join(d, "未来发展.png"))

        # 显示图像
        plt.imshow(wc, interpolation='bilinear')
        # interpolation='bilinear' 表示插值方法为双线性插值
        plt.axis("off")  # 关掉图像的坐标
        plt.show()

    if __name__ == '__main__':
        # 读取文件
        d = path.dirname(__file__)
        #  text 的具体内容（根据我们的 4 种分类更改）
        text = open(path.join(d, 'comments//1.txt'), 'r', encoding='utf-8').read()
        # 分词操作
        text = chnSegment.word_segment(text)
        # 生成词云
        WORLDCLOUD.generate_wordcloud(text)
```

代码 3 LDA 主题分析

```
import numpy as np
from gensim import corpora, models
import jieba
# 设置停用词，为哈工大停用词库中的所有词
stopwords
# 读取评论文本文件
text
# 未去掉停用词的分词结果 list 类型
text_split = jieba.cut(text)

# 去掉停用词的分词结果 list 类型
text_split_no = [word for word in text_split if word not in stopwords]
```

```
# 将去除停用词的分词结果写入文件，结束预处理
    fW.write(' '.join(text_split_no))

if __name__ == '__main__':
    # 读入文本数据
    f = open
    texts = [[word for word in line.split()] for line in f]
    f.close()
    M = len(texts)
    print('文本数目：%d 个' % M)
    # 建立词典
    dictionary = corpora.Dictionary(texts)
    V = len(dictionary)
    print('词的个数：%d 个' % V)
    # 计算文本向量
    corpus = [dictionary.doc2bow(text) for text in texts]
    # 计算文档 TF-IDF
    corpus_tfidf = models.TfidfModel(corpus)[corpus]
    # LDA 模型拟合
    num_topics = 5  # 定义主题数为 5
    lda = models.LdaModel(corpus_tfidf, num_topics=num_topics, id2word=dictionary,
                          alpha=0.01, eta=0.01, minimum_probability=0.001,
                          update_every=1, chunksize=100, passes=1)

    # 所有文档的主题
    doc_topic = [a for a in lda[corpus_tfidf]]
    print('Document-Topic:')
    print(doc_topic)
    # 每个主题的词分布
    num_show_term = 6  # 每个主题显示 6 个词
    for topic_id in range(num_topics):
        print('主题#%d：\t' % topic_id)
        term_distribute_all = lda.get_topic_terms(topicid=topic_id)  # 所有词的词分布
        term_distribute = term_distribute_all[:num_show_term]  # 只显示前几个词
        term_distribute = np.array(term_distribute)
        term_id = term_distribute[:, 0].astype(np.int)
        print('词：', end="")
        for t in term_id:
            print(dictionary.id2token[t], end=' ')
        print('概率：', end="")
        print(term_distribute[:, 1])

    # 将主题-词写入一个文档 topword.txt，每个主题显示 10 个词
    with open(r'D:\Users\26747\PycharmProjects\pythonProject2\正大杯\LDA 分析\主题合集\
零售业.txt', 'w', encoding='utf-8') as tw:
```

```
        for topic_id in range(num_topics):
            term_distribute_all = lda.get_topic_terms(topicid=topic_id, topn=10)
            term_distribute = np.array(term_distribute_all)
            term_id = term_distribute[:, 0].astype(np.int)
            for t in term_id:
                tw.write(dictionary.id2token[t] + " ")
            tw.write("\n")
```

代码 4 情感分析

```
from snownlp import SnowNLP

def emotionAnalysis(path):
    with open(path, encoding='utf-8') as f:
        text = [line.split(',')[-1] for line in f.readlines()[1:]]

    # 建立情感字典
    emotions = {
        'positive': 0,
        'negative': 0,
        'neutral': 0
    }

    for item in text:
        if SnowNLP(item).sentiments > 0.6:
            emotions['positive'] += 1
        elif SnowNLP(item).sentiments < 0.4:
            emotions['negative'] += 1
        else:
            emotions['neutral'] += 1

    print(emotions)

if __name__ == "__main__":
    path = r"D:\Users\26747\PycharmProjects\pythonProject2\正大杯\词云分析\comments\
零售业.txt"
    emotionAnalysis(path)
```

代码 5 熵权法代码

```
import numpy as np
import pandas as pd
from io import StringIO
```

```
# 数据字符串
data = """
Q27_1    Q27_2    Q27_3    Q27_4    Q27_5    Q27_6    Q27_7
具体数据
"""

# 将字符串转换为 DataFrame
data = StringIO(data)  # 将字符串转换为类文件对象
df = pd.read_csv(data, sep='\t')  # 使用制表符作为分隔符读取数据

# 熵权法计算权重
def entropy_weight_method(data):
    # 数据标准化（归一化）
    normalized_data = data / data.sum(axis=0)

    # 防止对 0 取对数
    normalized_data = normalized_data.replace(0, np.finfo(float).eps)

    # 计算每个指标的熵值
    entropy = -((normalized_data * np.log(normalized_data)).sum(axis=0) / np.log(len(data)))

    # 计算每个指标的差异系数
    diversity = 1 - entropy

    # 计算权重
    weights = diversity / diversity.sum()

    return weights

# 计算权重
weights = entropy_weight_method(df)
print("各指标权重：")
print(weights)

# 计算每个样本的综合得分
df['综合得分'] = df.dot(weights)
print("\n 综合得分：")
print(df)

# 将结果保存到 Excel 文件
output_file = 'results.xlsx'
df.to_excel(output_file, index=False)
print(f"结果已保存到 '{output_file}' 文件中。")
```

代码 6 MLP 算法

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.preprocessing import StandardScaler
import json
from io import StringIO
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.preprocessing import StandardScaler
# 上一问计算得到的数据
data = """
Q27_1    Q27_2    Q27_3    Q27_4    Q27_5    Q27_6    Q27_7    finalScore
"""

# 将字符串转换为 DataFrame
data = pd.read_csv(StringIO(data), sep='\t')

# 特征和目标变量
X = data.iloc[:, :-1]  # 第 1 列到倒数第二列为特征
y = data.iloc[:, -1]      # 最后一列为目标变量

# 检查特征数据是否为空或包含非数值数据
print("\n 特征数据 X：")
print(X)

# 数据标准化
scaler = StandardScaler()
try:
    X_scaled = scaler.fit_transform(X)
    print("\n 标准化后的特征数据 X_scaled：")
    print(X_scaled)
except Exception as e:
    print(f"\n 数据标准化失败：{e}")
    print("\nX 的数据类型：")
    print(X.dtypes)
    print("\nX 中是否存在非数值数据：")
    print(X.apply(lambda col: col.apply(type).value_counts()))

# 划分训练集、验证集和测试集（7:2:1）
```

```
X_train, X_temp, y_train, y_temp = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=1/3, random_state=42)

# 创建 MLP 模型
mlp = MLPRegressor(hidden_layer_sizes=(7, 7, 7), max_iter=1000, random_state=42)

# 训练模型
mlp.fit(X_train, y_train)

# 预测训练集、验证集和测试集
y_train_pred = mlp.predict(X_train)
y_val_pred = mlp.predict(X_val)
y_test_pred = mlp.predict(X_test)

# 计算相对误差
train_error = mean_absolute_percentage_error(y_train, y_train_pred)
val_error = mean_absolute_percentage_error(y_val, y_val_pred)
test_error = mean_absolute_percentage_error(y_test, y_test_pred)

print(f"训练集相对误差: {train_error:.4f}")
print(f"验证集相对误差: {val_error:.4f}")
print(f"测试集相对误差: {test_error:.4f}")

# 特征重要性排序（基于 MLP 的权重）
importances = np.abs(mlp.coefs_[0]).sum(axis=0)  # 第一层权重的绝对值之和
feature_importance = pd.Series(importances, index=X.columns)
feature_importance_sorted = feature_importance.sort_values(ascending=False)

# 输出绝对值排序
print("\n 特征重要性排序（绝对值）： ")
print(feature_importance_sorted)

# 归一化特征重要性
normalized_importance = feature_importance / feature_importance.sum()
normalized_importance_sorted = normalized_importance.sort_values(ascending=False)

# 输出归一化排序
print("\n 特征重要性排序（归一化）： ")
print(normalized_importance_sorted)

# 创建一个 DataFrame 保存结果
results_df = pd.DataFrame({
    "Feature": feature_importance_sorted.index,
    "Absolute Importance": feature_importance_sorted.values,
    "Normalized Importance": normalized_importance_sorted.values
```

```
})
#  保存到 Excel 文件
results_df.to_excel("feature_importance.xlsx", index=False)
print("\n 特征重要性结果已保存到  feature_importance.xlsx  文件中。")
```