

2024 睿抗信创大数据决赛（样题）--高职

样题说明：本样题，依据决赛竞赛规程中关于竞赛模块要求设计，仅供参考；

竞赛实操环境说明：
（切记：正式比赛，请您根据比赛环境的相关说明在服务器环境中搭建分布式集群环境，并填写相关答案；备赛情况，请依据题目要求自行搭建环境）

序号	配置项	详细信息
1	操作系统	中科方德服务器操作系统
2	HDFS	已安装
3	软件位置	/home/zkpk/soft
4	服务器	1 主节点（主机名：master ）
		2 计算节点：
		- 节点 1：slave01
		- 节点 2：slave02
5	用户名，密码	用户包含：root、zkpk；密码：zkpk

一、大数据平台搭建模块（20 分，共 20 个题目，每小题 1 分）

1. 在中科方德系统上搭建 HDFS 集群之前，需要在 master 节点上生成 SSH 密钥并配置免密登录，命令为：`_____`。
答案：ssh-keygen -t rsa
2. 在启动 HDFS 之前，需要在 master 节点上对 HDFS 进行格式化操作，使用的命令为：`_____`。
答案：hdfs namenode -format
3. 启动 HDFS 和 YARN 服务的命令分别为：`_____`和`_____`。
答案：start-dfs.sh 和 start-yarn.sh

4. 在中科方德系统中，为了赋予`zkpk`用户 sudo 权限，可以通过编辑`/etc/sudoers`文件，并添加以下内容：`_____ ALL=(ALL) NOPASSWD: ALL`。

答案：zkpk ALL=(ALL) NOPASSWD: ALL

5. 检查 HDFS 文件系统的运行状态（如剩余空间、集群健康状况等），使用的命令为：_____。

答案：`hadoop fs -report`

6.检查 HDFS 的当前状态和健康情况的命令为：_____。

答案：hdfs dfsadmin -report

7. 强制启动 HDFS 安全模式，可以使用的命令是：_____。

答案：hdfs dfsadmin -safemode enter

8. 在 Hive 的 hive-site.xml 文件中，配置连接元数据存储的 JDBC 驱动时，javax.jdo.option.ConnectionURL 的值应设置为 jdbc:mysql://master:3306/_____。

答案: hive

9. 启动 Hive 服务时，可以使用 hive --service _____ 命令启动 Hive Metastore 服务。

答案: metastore

10. 安装完 Hadoop 后，需将`mapred-site.xml.template`文件复制为`mapred-site.xml`，执行的命令为：`_____`。

答案：

Cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml

11. 在 hive-site.xml 中设置 Hive 临时目录的属性为 hive.exec.scratchdir，将该属性的值设置为 _____。

答案: /tmp/hive

12. HBase 依赖于 Zookeeper，因此在 hbase-env.sh 文件中，我们将 HBASE_MANAGES_ZK 设置为 _____，以禁用 HBase 自带的 Zookeeper 管理功能。

答案: false

13.启动 HBase 集群的命令是 _____，这将自动启动 HBase 及其内置的 Zookeeper。

答案: start-hbase.sh

14.在主节点上查看 HBase 是否启动，可以使用 jps 命令检查进程列表，确保显示 _____ 和 HRegionServer。

答案: HMaster

15. 在 HBase 中删除表之前，需要先禁用该表，可以使用命令 _____ 'test_table' 禁用表，然后用 drop 'test_table' 删除表。

答案: disable

16.为了与 Hive 集成，需要将 Hive 的 _____ 文件复制到 Spark 的 conf 目录中。

答案: hive-site.xml

17.在 Spark SQL 中，可以执行 SHOW TABLES; 命令来查看集成的 Hive 表，随后可以使用 _____ * FROM your_hive_table LIMIT 10; 来查询表内容。

答案: SELECT

18.在 Spark Shell 中，可以使用 sc.parallelize 函数创建一个并行集合，比如 val data = sc.parallelize(1 to 1000)，然后通过 _____ 方法计算元素数量。

答案: count()

19.启动 Spark Master 服务的命令是 _____，该命令会启动 Spark 主节点。

答案: start-master.sh

20.如果 Spark 使用 MariaDB 作为 Hive 的元数据存储，需要确保 MariaDB 的 JDBC 驱动文件位于 Spark 的 _____ 目录下。

答案: jars

二、离线数据处理（30 分，每空 3 分，共 10 题）

单词计数项目

1.项目背景描述：

本项目旨在通过离线数据分析实现对大规模文本数据的词频统计。具体应用场景可能包括文档的内容分析、文本挖掘、信息检索以及自然语言处理中的基础处理步骤。由于数据量较大且需支持高并发的分布式计算，项目基于 HDFS 集群存储文件，并借助 Hadoop 生态系统的其他组件（如 Hive 和 Spark）进行计算处理，实现单词计数任务。

2.涉及主要技术

1. **HDFS (Hadoop Distributed File System)**：提供分布式存储，用于保存大规模的文本文件数据。
2. **Hive**：数据仓库工具，用于数据查询和分析，提供 SQL 风格的查询功能。
3. **ZooKeeper**：提供分布式协调服务，保证数据一致性。
4. **HBase**：NoSQL 数据库，存储和管理结构化数据，适合高并发和快速数据读写的场景。
5. **Spark**：分布式计算框架，适合大规模数据的快速处理，支持离线数据分析。
6. **Python**：作为编程语言进行数据处理和分析。
7. **PyCharm**：集成开发环境（IDE），用于 Python 编程开发，提高开发效率。

3.word.csv 文件单词情况

- 文件名：`word.csv`
- 文件描述：包含 2000 个单词的 CSV 文件，每行记录一个单词，可能存在重复的单词。
- 数据示例：

```
word1
word2
```

word3

...
...

文件中的单词数量较多，便于进行大规模词频统计。

4.项目目录结构

以下是项目的建议目录结构：

```
...  
word_count_project/  
|  
├── data/  
|   └── word.csv                # 原始数据文件  
|  
├── src/  
|   ├── __init__.py            # 初始化文件  
|   ├── data_loader.py         # 数据加载模块  
|   ├── word_count.py          # 词频统计主模块  
|   ├── hive_integration.py     # Hive 查询与数据整合模块  
|   └── hdfs_operations.py      # HDFS 读写操作模块  
|  
├── config/  
|   └── config.yaml            # 配置文件（如 HDFS、Hive、Spark 等相关配置）  
|  
├── results/  
|   └── word_count_output.csv   # 输出结果文件（包含每个单词的计数结果）  
|  
├── logs/  
|   └── word_count.log          # 运行日志  
|  
└── README.md                  # 项目说明文档  
...
```

5.项目主要功能模块说明

- 1. ****data_loader.py****: 加载 `word.csv` 中的单词数据并准备数据输入。
- 2. ****word_count.py****: 实现词频统计核心逻辑，将单词进行计数并保存结果。

3. **hive_integration.py**: 利用 Hive 对数据进行 SQL 查询操作, 进一步进行数据分析。

4. **hdfs_operations.py**: 对 HDFS 文件系统中的数据进行操作, 如上传和读取 `word.csv`。

6. 注意事项

1. 需要确保 HDFS 和 Hive 配置正确, 且已启动服务。
2. 使用 `pyhive` 和 `hdfs` 库时, 需提前安装依赖项
3. 根据实际的 HDFS 路径和 Hive 配置调整代码中的参数。

7. 依据上面的项目背景在 pycharm 开发环境中开发项目, 并完成如下题目:

1. 在 `data_loader.py` 中, 加载 CSV 文件的函数 `load_data` 使用了 `pandas` 库来读取文件。请补全以下代码以完成 CSV 文件的加载:

```
```python
def load_data(file_path):
 data = pd.read_csv(_____)
 words = data[0].tolist()
 return words
```
```

>答案: `file_path`

2. 在 `word_count.py` 中, `count_words` 函数用于计算单词频率。请填空, 补全代码以完成词频统计:

```
```python
from collections import _____

def count_words(words):
 word_count = _____(words)
 return word_count
```
```

>答案: `Counter`, `Counter`

3. 在 `word_count.py` 中, `save_results` 函数用于将词频统计结果保存到 CSV 文件中。请补全以下代码以指定列名:

```
```python
def save_results(word_count, output_file):
 df = pd.DataFrame(word_count.items(), columns=[_____, "count"])
 df.to_csv(output_file, index=False)
```
```

>答案: `"word"`

4. 在 `hive_integration.py` 中, `connect_hive` 函数用于连接 Hive 数据库。请补全以下代

码，以指定默认的 Hive 主机和端口：

```
```python
def connect_hive(host=_____, port=10000, database='default'):
 conn = hive.Connection(host=host, port=port, database=database)
 return conn
```
```

>答案：`localhost`

5. 在 `hive_integration.py` 中，`query_hive` 函数用于在 Hive 中执行 SQL 查询。请填空补全代码，以获取查询结果：

```
```python
def query_hive(conn, query):
 cursor = conn._____
 cursor.execute(query)
 return cursor._____
```
```

>答案：`cursor()`, `fetchall()`

6. 在 `hdfs_operations.py` 中，`upload_to_hdfs` 函数用于将本地文件上传到 HDFS。请补全以下代码以实现文件上传：

```
```python
client = InsecureClient(_____)
client.upload(hdfs_path, _____)
```
```

>答案：`hdfs_url`, `local_path`

7. 在 `hdfs_operations.py` 中，`read_from_hdfs` 函数从 HDFS 读取文件内容。请填空以创建读取对象：

```
```python
with client.read(_____) as reader:
 return reader._____
```
```

>答案：`hdfs_path`, `read()`

8. 在主脚本中，将单词计数结果保存到本地 CSV 文件后，通过 `upload_to_hdfs` 函数上传结果。请填空补全代码：

```
```python
save_results(word_count, 'results/word_count_output.csv')
upload_to_hdfs(_____, '/user/zkpk/word_count_output.csv')
```
```

>答案：`results/word_count_output.csv`

9. 在主脚本中，通过 `connect_hive` 函数建立与 Hive 的连接。请填空补全代码以完成连接：

```
```python
conn = connect_hive()
if conn:
 query_result = query_hive(conn, _____)
 print("Query result:", query_result)
```
```

>答案: ``SELECT * FROM word_count_table LIMIT 10``

10. `data_loader.py` 中, `load_data` 函数返回单词列表。请填空, 以便在 `word_count.py` 中调用该函数并完成单词列表加载:

```
```python
from data_loader import _____

words = _____('data/word.csv')
```
```

>答案: `load_data`, `load_data`

三、数据分析挖掘 (20 分, 每题 2 分, 共 10 题)

分析消费者行为, 实现市场细分和精准广告推送。

1.项目背景描述

本项目旨在通过数据分析和挖掘技术对客户进行细分, 以实现精准的营销策略。随着市场竞争的加剧, 企业面临如何有效触达和服务不同类型消费者的挑战。基于消费者行为数据的深入分析, 我们可以识别不同的客户群体特征和行为模式, 进而实现市场细分。这不仅能帮助企业在有限的预算内优化广告投放, 还能提升客户体验和满意度。

通过分析消费者的购买记录、浏览行为、偏好和消费习惯等多维数据, 项目将生成客户的细分模型, 并设计出个性化的营销方案, 以实现精准广告推送。最终目标是提升营销效率, 增加客户转化率, 为企业带来更高的投资回报率(ROI)。

2.项目目录结构

```
customer_segmentation_marketing/
|
├── data/                                # 数据文件夹
|   ├── raw_data.csv                    # 原始客户行为数据
|   ├── processed_data.csv              # 预处理后的数据
|   └── segmentation_model.pkl          # 客户细分模型文件
```

| | | |
|--|------------------------------|-------------------------------|
| | | |
| | — src/ | # 代码文件夹 |
| | — __init__.py | # 初始化文件 |
| | — data_preprocessing.py | # 数据预处理模块 |
| | — segmentation_model.py | # 客户细分模型模块 |
| | — behavior_analysis.py | # 行为分析模块 |
| | — marketing_strategy.py | # 营销策略生成模块 |
| | — evaluation.py | # 效果评估模块 |
| | — config/ | # 配置文件夹 |
| | — config.yaml | # 配置文件（如数据库连接、模型参数等） |
| | — results/ | # 分析和模型结果文件夹 |
| | — segmentation_report.csv | # 客户细分报告 |
| | — marketing_results.csv | # 营销策略结果 |
| | — evaluation_metrics.csv | # 效果评估指标 |
| | — logs/ | # 日志文件夹 |
| | — project.log | # 项目运行日志 |
| | — notebooks/ | # Jupyter notebooks 用于探索性数据分析 |
| | — data_exploration.ipynb | # 数据探索 notebook |
| | — tests/ | # 测试文件夹 |
| | — __init__.py | # 初始化文件 |
| | — test_data_preprocessing.py | # 测试数据预处理模块 |
| | — test_segmentation_model.py | # 测试客户细分模型模块 |
| | — test_behavior_analysis.py | # 测试行为分析模块 |
| | — test_marketing_strategy.py | # 测试营销策略生成模块 |
| | — README.md | # 项目说明文档 |

3.主要模块说明

1. **data_preprocessing.py**: 负责加载原始数据、数据清洗、特征工程等预处理步骤，为后续分析提供标准化数据。
2. **segmentation_model.py**: 实现客户细分模型的构建与训练，例如 KMeans 聚类或分层聚类，保存模型以便后续预测。
3. **behavior_analysis.py**: 分析客户行为模式，计算关键行为指标（如购买频率、消费金额等），为客户分群提供数据支持。
4. **marketing_strategy.py**: 基于客户细分结果生成个性化的营销策略，包括广告推荐、优惠推送等。

5. **evaluation.py**: 对客户细分与营销效果进行评估，生成各项指标（如 ROI、客户转化率）以判断策略的有效性。

4.其他文件说明

- **config.yaml**: 存放项目的参数配置（如数据库路径、模型参数等），便于管理。
- **project.log**: 记录项目的运行日志，包括数据处理、模型训练和策略生成的过程。
- **data_exploration.ipynb**: 用 Jupyter Notebook 进行数据探索和可视化分析，帮助深入了解客户数据特征。
- **tests/**: 用于各模块的单元测试，确保代码功能的正确性和稳定性。

5.依据上面的项目背景在 **pycharm** 开发环境中开发项目，并完成如下题目：

1. 在 `data_preprocessing.py` 中，`preprocess_data` 函数中使用 `StandardScaler` 对特征进行标准化。请填空补全代码以实现标准化处理：

```
from sklearn.preprocessing import _____

scaler = _____()

features = data[['purchase_frequency', 'average_spending', 'last_purchase_days']]

data[['purchase_frequency', 'average_spending', 'last_purchase_days']] =
scaler.fit_transform(features)
```

答案：StandardScaler, StandardScaler

2. 在 `segmentation_model.py` 中，`train_segmentation_model` 函数中使用 `KMeans` 进行聚类。请填空补全代码以训练模型：

```
from sklearn.cluster import _____

model = _____(n_clusters=n_clusters, random_state=42)

segments = model.fit_predict(data[['purchase_frequency', 'average_spending',
'last_purchase_days']])
```

填空答案：KMeans, KMeans

3. 在 `segmentation_model.py` 中，`save_model` 函数用于将模型保存到文件。请填空补全代码：

```
with open(file_path, 'wb') as f:  
  
    pickle._____(model, f)
```

填空答案：dump

4. 在 `segmentation_model.py` 中，`load_model` 函数用于加载保存的模型。请填空补全代码：

Python

```
with open(file_path, 'rb') as f:  
  
    model = pickle._____(f)
```

填空答案：load

5. 在 `behavior_analysis.py` 中，`analyze_behavior` 函数中使用 `groupby` 方法计算每个客户群体的行为统计。请填空补全代码：

```
segment_analysis = data.groupby('_____').agg({  
  
    'purchase_frequency': 'mean',  
  
    'average_spending': 'mean',  
  
    'last_purchase_days': 'mean'  
  
}).reset_index()
```

填空答案：segment

6. 在 `marketing_strategy.py` 中，`generate_marketing_strategy` 函数根据 `segment_analysis` 的结果生成策略。请填空补全代码：

```
for _, row in segment_analysis.iterrows():  
  
    if row['average_spending'] > 1.0 and row['purchase_frequency'] > 1.0:  
  
        strategy[row['segment']] = "优先推送高价产品和奖励积分"  
  
    elif row['_____'] < 0.5:  
  
        strategy[row['segment']] = "提供近期购买的优惠券"
```

填空答案: last_purchase_days

7. 在 `evaluation.py` 中, `evaluate_segmentation` 函数中使用 `silhouette_score` 评估聚类效果。请填空补全代码:

```
from sklearn.metrics import _____  
  
score = _____(data[['purchase_frequency', 'average_spending',  
                           'last_purchase_days']], labels)
```

填空答案: silhouette_score, silhouette_score

8. 在 `evaluation.py` 中, `evaluate_marketing_strategy` 函数用于计算营销策略的转化率。请填空补全代码:

```
converted_customers = len(data[data['_____'] == True])  
  
conversion_rate = converted_customers / total_customers
```

填空答案: converted

9. 在主脚本中, 加载并预处理数据时, 需要调用 `preprocess_data` 函数。请填空补全代码:

```
from data_preprocessing import load_data, preprocess_data

data = load_data('data/raw_data.csv')

preprocessed_data = preprocess_data(_____)
```

填空答案：data

10. 在主脚本中，调用 `analyze_behavior` 函数以生成客户群体的行为统计。请填空补全代码：

Python

```
from behavior_analysis import analyze_behavior

segment_analysis = _____(segmented_data)
```

填空答案：analyze_behavior

四、数据可视化（15 分，共 1 题）

“智慧学习可视化分析”

1.项目背景描述

随着教育信息化的不断发展，基于大数据的智慧学习逐渐成为现代教育的核心组成部分。智慧学习结合了数据分析、机器学习和人工智能技术，通过对学习行为、学习成绩、学习习惯等数据的分析，能够实现个性化的学习路径和学习策略优化，为学生提供更为精准的学习建议。

在本项目中，智慧学习系统将使用大规模教育数据来挖掘学生的学习模式和行为偏好，分析不同学习阶段和学习风格下的效果差异。这些数据为教师和教育管理者提供了科学依据，能够帮助他们制定合理的教学方案，提升教育质量。

2.数据分析与可视化

我们有以下数据样例，用于展示学生在学习活动上的时间分布情况：

| 学习活动 | 时间 (小时) |
|------|---------|
| 阅读 | 50 |
| 练习 | 30 |
| 视频学习 | 40 |
| 测验 | 20 |
| 项目实践 | 60 |

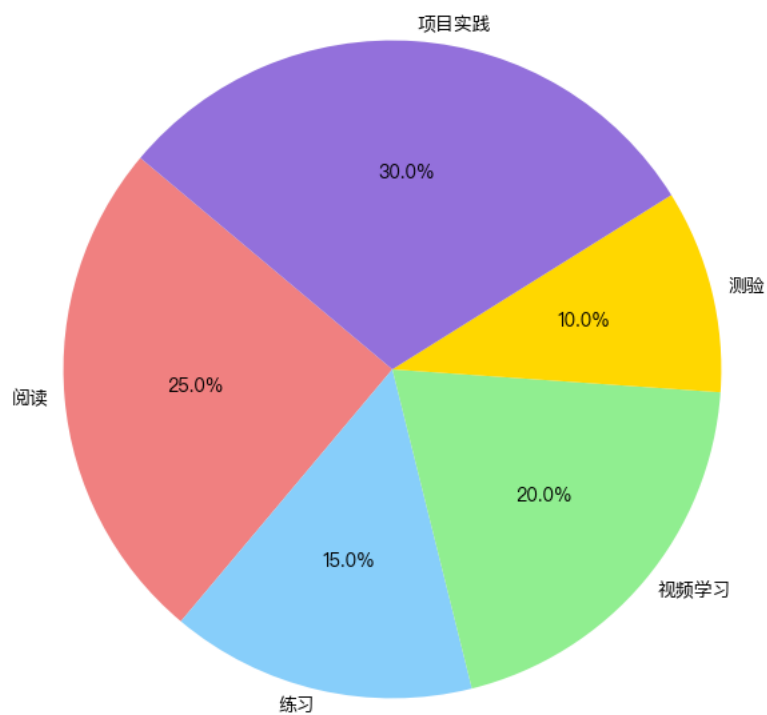
题目要求：

请您依据上述背景和给出的数据，使用 Python 和 Matplotlib 创建柱状图和饼状图，表达“学习活动时间比例”、“ 学生在不同学习活动上的时间分布”， 请你根据理解， 来对展示结果进行分析与说明（备注： 尽量分条目描述， 1-5 条）

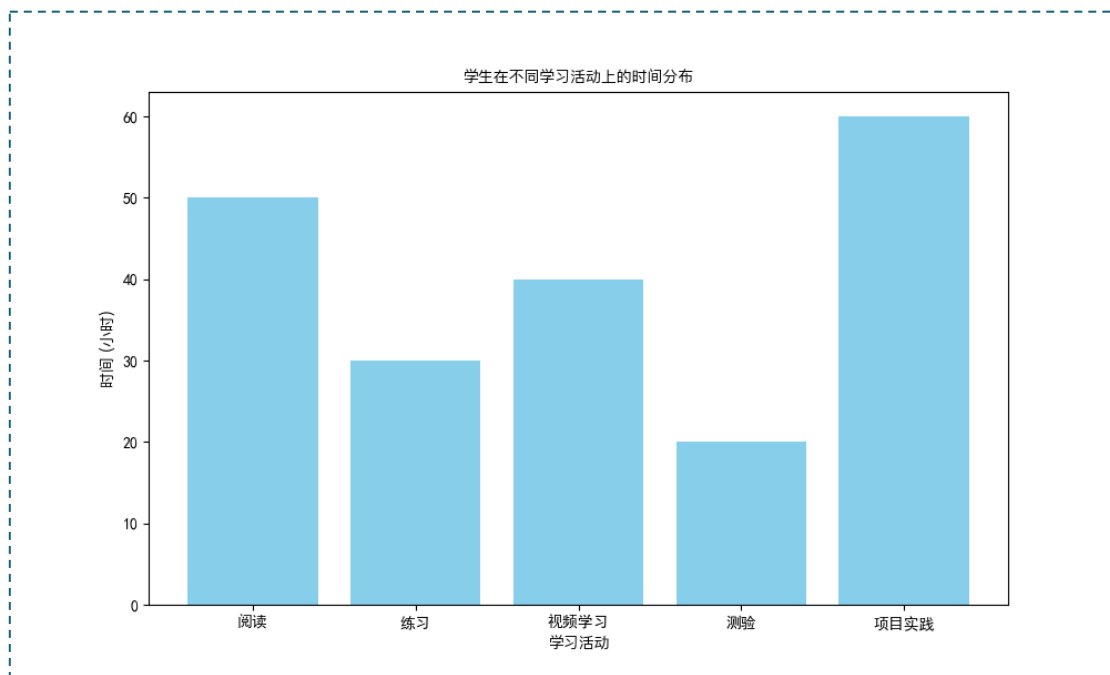
（正式比赛会给出相关数据，且仅有数据，备考情况，请自行搭建环境）

参考答案：

学习活动时间分布比例



学习活动时间比例



学生不同学习活动上的时间分布

学习活动时间比例

1. 饼状图显示，项目实践占据了总学习时间的最大比例，达到了 31.6%，是最耗时的活动。
2. 阅读和视频学习也占据了较大比例，分别为 26.3%和 21.1%，说明学生在获取信息和知识时倾向于利用这类活动。
3. 练习和测验的时间比例相对较小，占总时间的 15.8%和 10.5%，这表明在实际测试和强化训练方面的时间较少。
4. 各活动的比例展示了学生学习行为的侧重点，实践与自学显然成为主流。
5. 此比例分析有助于教育者了解学生在学习活动中的时间分配，从而在教学设计中平衡不同活动类型。

学生不同学习活动上的时间分布

1. 从柱状图可以看出，学生在项目实践上的时间投入最多，达到 60 小时，远高于其他活动。
2. 阅读和视频学习也占据了较多的时间，分别为 50 小时和 40 小时，显示出学生对这些活动的重视。
3. 相比之下，学生在练习和测验上的时间较少，尤其是测验时间，仅有 20 小时。
4. 整体来看，时间分布呈现出向项目实践、阅读等活动倾斜的趋势，可能反映了学生对动手实践和自主学习的偏好。
5. 不同活动间的时间差异显示出学习活动的多样化需求，有助于了解学生的学习偏好以制定相应的教学策略。

五、综合分析（10 分，共 1 题）

商超市场分析与智慧营销

商超的行业背景描述

北京某大型商超面临复杂的市场分析需求，希望通过对大数据的深度挖掘，获得关于消费者行为、市场细分及竞争对手动态的洞察，从而优化商品陈列和营销策略。商超不仅需要了解消费者的购买偏好和购物频率，还要识别不同消费群体的特征，制定精准的营销活动。同时，监测竞争对手的促销动态和市场表现也至关重要，以便快速响应市场变化，保持竞争优势。

具体而言，商超的主要问题包括：

- 消费者行为识别：**了解顾客在不同商品类别上的消费行为，如高频商品、季节性商品和促销商品的需求波动。
- 精准市场细分：**针对家庭主妇、上班族、年轻人等不同群体，识别他们的购物习惯并推送有针对性的优惠信息。
- 竞争动态跟踪：**通过监测其他商超的促销活动、特定商品的折扣率和价格变化，优化自有商超的定价和促销策略。
- 实时数据处理：**有效分析快速变化的销售数据，调整库存和补货，避免商品短缺或库存积压。

请您依据上述行业背景分析，从技术视角给出合理的解决方案；

参考答案：

使用大数据技术的解决方案

为解决上述问题，北京某商超可以利用大数据技术打造一个市场分析系统，实现消费者行为洞察、市场细分和竞争动态跟踪。以下是该方案的具体技术思路：

1. 数据收集与存储

- 数据源：**通过 POS 系统、会员管理系统、APP 以及微信公众号等收集销售、会员和购物行为数据，了解消费者在不同品类商品上的购买行为。
- 数据存储：**利用 HDFS 进行分布式存储，保障商超每日产生的海量数据可以安全高效地存储并快速读取。

2. 数据预处理

- 清洗和整合：**使用 ETL 工具清洗数据，去除重复记录并处理缺失值，确保数据的完整性和准确性。

- **特征提取：**从交易数据中提取商品类别、消费金额、购买频率、会员等级等特征，为后续分析提供数据支持。

3. 消费者行为分析

- **数据分析和模式识别：**使用 Spark、Hadoop 等分布式处理工具，对顾客的消费习惯进行分析，识别高频购买的商品、季节性商品需求以及促销商品的销售趋势。
- **机器学习模型：**使用聚类算法（如 KMeans、DBSCAN）进行市场细分，区分不同消费群体并构建用户画像，为定制化营销活动提供依据。

4. 竞争对手分析

- **情报收集：**从公开渠道（如社交媒体、竞争对手的网站、价格比较平台）获取竞品信息，了解其他商超的促销活动和价格变化。
- **NLP 分析：**利用自然语言处理技术，对文本信息进行情感分析和关键词提取，帮助分析竞争对手的市场策略和产品优劣。
- **市场趋势预测：**基于时间序列模型预测特定商品的价格波动，为商超提供竞争性定价建议。

5. 数据可视化

- **数据可视化工具：**使用 Tableau 或 Power BI 创建实时可视化面板，显示消费者偏好、促销效果和竞争动态。
- **自定义仪表盘：**展示关键绩效指标（KPI），如日均销量、库存水平、促销效果等，以便管理者快速了解业务情况。
- **自动报告生成：**定期生成市场分析报告，包含销售趋势、市场份额和顾客反馈，以支持商超的决策。

6. 技术架构概述

1. **数据层：**HDFS 存储所有数据，Kafka 处理实时数据流，Hive 和 Spark SQL 提供查询支持。
2. **分析层：**Spark 负责批量处理和实时数据分析，使用 Scikit-learn 和 TensorFlow 进行市场细分和行为预测。
3. **展示层：**Tableau 和 Power BI 实现数据可视化，帮助决策者随时掌握市场动态和消费者需求。

总结

通过大数据技术，商超可以构建一个智能市场分析系统，深入挖掘消费者行为、有效实施市场细分和及时跟踪竞争动态，从而提升业务效率、优化库存管理，并提高整体市场竞争力。

六、职业素养（5分）

（略）

职业素养模块评分标准

1. **沟通表达**：选手是否具备良好的语言表达能力，能否流畅、清晰地进行信息传达，展现出理解和反馈的能力。
2. **团队合作**：选手是否能够与团队成员紧密合作，是否能在团队中发挥积极作用。
3. **时间管理**：考察选手的时间观念，是否按时完成任务，是否能够高效利用时间。
4. **职业态度**：选手的责任心和工作积极性，是否表现出较高的职业热情。
5. **应急处理能力**：面对问题或突发情况时，选手是否具备灵活应对、沉着处理的能力。