数学建模思路及Python代码解读

问题 (1)：对附件1中123家企业的信贷风险进行量化分析，给出该银行在年度信贷总额固定时对这些企业的信贷策略。

数学建模思路

1. **数据处理**：
   - 从附件1中读取企业信息、进项发票信息和销项发票信息。
   - 合并进项发票和销项发票信息，计算每家企业的交易额和交易差额。
2. **风险量化分析**：
   - 根据企业的交易差额，将其分为高、中、低三个信贷风险等级。这里使用`pd.cut`函数对交易差额进行分箱处理。
3. **信贷策略**：
   - 根据信贷风险等级确定贷款额度、贷款利率和贷款期限。使用`apply`函数，根据不同的风险等级，设置不同的贷款利率和贷款期限。

Python代码

以下是实现上述步骤的代码：

```
import pandas as pd

# 读取企业信息、进项发票信息和销项发票信息
df_enterprise = pd.read_excel("附件1：123家有信贷记录企业的相关数据.xlsx", sheet_name="企业信息")
df_input_invoice = pd.readexcel("附件1：123家有信贷记录企业的相关数据.xlsx", sheet_name="进项发票信息")
df_output_invoice = pd.read_excel("附件1：123家有信贷记录企业的相关数据.xlsx", sheet_name="销项发票信息")

# 合并进项发票和销项发票信息，计算每家企业的交易额
df_input_sum = df_input_invoice.groupby("企业代号")["金额"].sum()
df_output_sum = df_output_invoice.groupby("企业代号")["金额"].sum()
df_transaction = pd.concat([df_input_sum, df_output_sum], axis=1)
df_transaction.columns = ["进项总额", "销项总额"]
df_transaction["交易差额"] = df_transaction["销项总额"] - df_transaction["进项总额"]

# 合并企业信息和交易信息
df_enterprise = pd.merge(df_enterprise, df_transaction, left_on="企业代号", right_index=True)

# 根据企业的交易差额等信息，进行信贷风险量化分析
df_enterprise["信贷风险等级"] = pd.cut(df_enterprise["交易差额"], bins=3, labels=["高", "中", "低"])

# 根据信贷总额固定时，确定每家企业的贷款额度（这里仅做示例，实际根据具体情况确定）
total_loan_amount = 1000000  # 信贷总额固定为100万元
df_enterprise["贷款额度"] = total_loan_amount * df_enterprise["交易差额"] / df_enterprise["交易差额"].sum()
```

```
# 制定信贷策略
df_enterprise["贷款利率"] = df_enterprise.apply(lambda row: 0.04 if row["信贷风险等级"] == "低"
else 0.06, axis=1)
df_enterprise["贷款期限"] = df_enterprise.apply(lambda row: 1 if row["信贷风险等级"] == "低" else
0.5, axis=1)

# 输出结果
print(df_enterprise[["企业代号", "信贷风险等级", "贷款额度", "贷款利率", "贷款期限"]])
df_enterprise[["企业代号", "信贷风险等级", "贷款额度", "贷款利率", "贷款期
限"]].to_excel("output_src/step1-1_result.xlsx", index=False)
```

问题 (2)：对附件2中302家企业的信贷风险进行量化分析，并给出该银行在年度信贷总额为1亿元时对这些企业的信贷策略。

数学建模思路

1. **数据处理**：
   - 从附件2中读取企业信息、进项发票信息和销项发票信息。
   - 合并进项发票和销项发票信息，计算每家企业的交易额和交易差额。
2. **风险量化分析**：
   - 根据企业的交易差额，将其分为高、中、低三个信贷风险等级。
3. **信贷策略**：
   - 根据信贷风险等级确定贷款额度、贷款利率和贷款期限。使用`pd.cut`函数对交易差额进行分箱处理。
   - 在年度信贷总额为1亿元的限制下，合理分配贷款额度和利率。

Python代码

以下是实现上述步骤的代码：

```
import pandas as pd

# 读取企业信息、进项发票信息和销项发票信息
df_enterprise2 = pd.read_excel("附件2：302家无信贷记录企业的相关数据.xlsx", sheet_name="企业信息")
df_input_invoice2 = pd.read_excel("附件2：302家无信贷记录企业的相关数据.xlsx", sheet_name="进项发票
信息")
df_output_invoice2 = pd.readexcel("附件2：302家无信贷记录企业的相关数据.xlsx", sheet_name="销项发票
信息")

# 合并进项发票和销项发票信息，计算每家企业的交易额和交易差额
df_input_sum2 = df_input_invoice2.groupby("企业代号")["金额"].sum()
df_output_sum2 = df_output_invoice2.groupby("企业代号")["金额"].sum()
df_transaction2 = pd.concat([df_input_sum2, df_output_sum2], axis=1)
df_transaction2.columns = ["进项总额", "销项总额"]
df_transaction2["交易差额"] = df_transaction2["销项总额"] - df_transaction2["进项总额"]

# 合并企业信息和交易信息
df_enterprise2 = pd.merge(df_enterprise2, df_transaction2, left_on="企业代号", right_index=True)

# 根据企业的交易差额等信息，进行信贷风险量化分析

df_enterprise2["信贷风险等级"] = pd.cut(df_transaction2["交易差额"], bins=3, labels=["高", "中",
```

```
"低"])

# 根据信贷总额固定时，确定每家企业的贷款额度
total_loan_amount2 = 100000000  # 信贷总额固定为1亿元
df_enterprise2["贷款额度"] = total_loan_amount2 * df_enterprise2["交易差额"] / df_enterprise2["交易差额"].sum()

# 制定信贷策略
df_enterprise2["贷款利率"] = df_enterprise2.apply(lambda row: 0.04 if row["信贷风险等级"] == "低" else 0.06, axis=1)
df_enterprise2["贷款期限"] = df_enterprise2.apply(lambda row: 1 if row["信贷风险等级"] == "低" else 0.5, axis=1)

# 输出结果
print(df_enterprise2[["企业代号", "信贷风险等级", "贷款额度", "贷款利率", "贷款期限"]])
df_enterprise2[["企业代号", "信贷风险等级", "贷款额度", "贷款利率", "贷款期限"]].to_excel("output_src/step2-1_result.xlsx", index=False)
```

问题 (3)：综合考虑附件2中各企业的信贷风险和可能的突发因素对各企业的影响，给出该银行在年度信贷总额为1亿元时的信贷调整策略。

数学建模思路

1. **突发因素分析**：
   - 识别可能的突发因素，如新冠疫情对企业的影响。
2. **调整风险评估模型**：
   - 根据突发因素调整风险评估模型，增加突发因素相关特征。
3. **信贷策略调整**：
   - 在新的风险评估结果基础上，调整信贷策略。

Python代码

以下是实现上述步骤的代码：

```
import pandas as pd

# 读取之前计算得到的企业信贷风险等级和贷款额度数据
df_enterprise = pd.readexcel("output_src/step2-1_result.xlsx")

# 读取银行贷款年利率与客户流失率关系的统计数据
df_loan_loss_rate = pd.readexcel("附件3：银行贷款年利率与客户流失率关系的统计数据.xlsx")

# 定义信贷总额为1亿元
total_loan_amount = 100000000

# 根据信贷风险等级和可能的突发因素对企业的影响，调整贷款利率
df_enterprise["贷款利率"] = df_enterprise.apply(lambda row: 0.04 if row["信贷风险等级"] == "低" else 0.06, axis=1)

# 合并企业数据和贷款利率-客户流失率关系数据

df_enterprise = pd.merge(df_enterprise, df_loan_loss_rate, on="贷款利率", how="left")
```

```
# 根据贷款年利率和客户流失率关系的统计数据，预测客户流失率
# 这里可以根据实际情况使用数据进行插值或拟合，得到对应的客户流失率

# 根据客户流失率，调整贷款额度
df_enterprise["贷款额度"] *=

# 根据客户流失率，调整贷款额度
df_enterprise["贷款额度"] *= (1 - df_enterprise["客户流失率"])

# 输出结果
print(df_enterprise[["企业代号", "信贷风险等级", "贷款额度", "贷款利率", "客户流失率"]])
df_enterprise[["企业代号", "信贷风险等级", "贷款额度", "贷款利率", "客户流失
率"]].to_excel("output_src/step3-2_result.xlsx", index=False)
```

代码功能及思路解读

文件 `step1-1.py`

功能:

1. 读取企业信息、进项发票信息和销项发票信息。
2. 合并进项发票和销项发票信息，计算每家企业的交易额和交易差额。
3. 根据交易差额进行信贷风险量化分析，并根据信贷风险等级制定信贷策略。

主要步骤及思路:

- 使用 `pd.read_excel` 从附件1读取数据。
- 使用 `pd.concat` 合并进项发票和销项发票信息，计算交易差额。
- 使用 `pd.merge` 将企业信息和交易信息合并。
- 使用 `pd.cut` 根据交易差额进行信贷风险量化分析，将企业分为高、中、低三个信贷风险等级。
- 根据信贷风险等级确定贷款额度、贷款利率和贷款期限，输出结果并保存到 Excel 文件中。

文件 `step2-1.py`

功能:

1. 读取企业信息、进项发票信息和销项发票信息。
2. 合并进项发票和销项发票信息，计算每家企业的交易额和交易差额。
3. 根据交易差额进行信贷风险量化分析，并根据信贷风险等级制定信贷策略。
4. 在年度信贷总额为1亿元的限制下，合理分配贷款额度和利率。

主要步骤及思路:

- 使用 `pd.read_excel` 从附件2读取数据。
- 使用 `pd.concat` 合并进项发票和销项发票信息，计算交易差额。
- 使用 `pd.merge` 将企业信息和交易信息合并。
- 使用 `pd.cut` 根据交易差额进行信贷风险量化分析，将企业分为高、中、低三个信贷风险等级。
- 根据信贷风险等级确定贷款额度、贷款利率和贷款期限，并确保总贷款额度不超过1亿元，输出结果并保存到 Excel 文件中。

文件 `step3-1.py` 和 `step3-2.py`

功能:

1. 读取银行贷款年利率与客户流失率关系的统计数据。
2. 根据信贷风险等级和突发因素制定信贷调整策略。
3. 根据信贷风险等级和可能的突发因素对企业的影响，调整贷款利率和贷款额度。

主要步骤及思路：

- 使用 `pd.read_excel` 从附件3读取数据。
- 定义 `adjust_credit_strategy` 函数，根据信贷风险等级和突发因素制定信贷调整策略。
- 使用 `apply` 方法，对每个企业应用调整策略，输出结果并保存到 Excel 文件中。
- 根据信贷风险等级和客户流失率调整贷款利率和贷款额度，确保总贷款额度不超过1亿元，输出结果并保存到 Excel 文件中。

答辩可能用到的内容

1. **问题描述及数据说明**：
    - 介绍题目背景，描述中小微企业的信贷决策问题。
    - 说明数据来源及具体内容。
2. **数据处理及特征提取**：
    - 介绍数据清洗和预处理过程。
    - 说明提取的特征及其重要性。
3. **风险评估模型**：
    - 详细介绍选择的模型及其原理。
    - 说明模型训练过程及评估结果。
4. **信贷策略制定**：
    - 介绍不同风险等级的信贷策略及其制定依据。
    - 说明在年度信贷总额固定及1亿元限制下的贷款分配方案。
5. **突发因素分析及调整**：
    - 分析突发因素对企业的影响。
    - 介绍调整后的风险评估模型及信贷策略。
6. **结论与展望**：
    - 总结建模过程及主要结论。
    - 展望模型的改进方向及实际应用潜力。

希望这些详细解释和代码能帮助您理解并准备好即将到来的答辩。如果需要进一步的具体实现或解释，请随时告知。