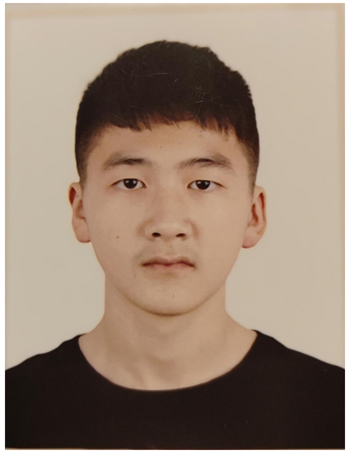
➢ Personal Introduction



香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

RED BIRD CHALLENGE CAMP

PORTFOLIO FOR HKUST(GZ)

YIXIAO ZHANG
Chengdu, China
+86 18202429280
q18202429280@outlook.com
linkedin.com/in/yixiao-zhang-2042ab2a3/
github.com/13558882230

Northeastern University
Bachelor of Management
Major:Information Management & Information Systems
Sep 2022 - Jul 2026 | GPA: 3.47/5 (Top 25%)

Hanyang University
Hanyang International Winter School
GPA: 4.25/4.5 (Top 5%)

Data Analysis & Visualization: Proficient in Python, R, Jupyter Notebook for data mining; skilled in Tableau and Streamlit for interactive visualizations.

Full-Stack Development: Backend: Flask framework for stable service development;Frontend: Angular, HTML/CSS/JavaScript for responsive interface design.

Database Management: MySQL (relational) and Elasticsearch (vectorized) .

Tools & Infrastructure: Git (version control), VS Code, Linux, Docker (containerization).

Programming Languages: Python, Java, R, SQL.

Language Proficiency: CET-6 (English); Mandarin Level 2-B.

HONORS AND AWARDS

Business Elite Challenge Accounting and Business Case Competition(**National First Prize)**
Northeastern University 11th Overseas Economic Management Scholars Seminar(**Outstanding Camper**）
Estonian National Summer School Full Scholarship (2025) (17/500)   €1,050 scholarship by Estonian Education and Youth Board.
Bucharest Summer University Partial Scholarship (2025)   Selected for partial funding at 19th BSU (750+ participants).

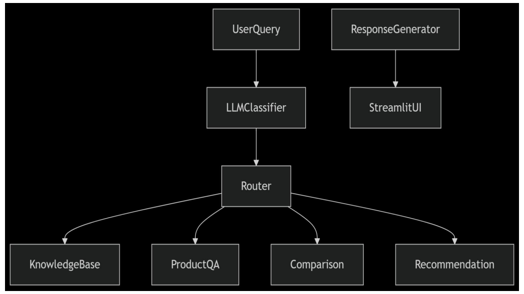
➢ Work experience

Xiaoduo Tech (7-9 2024) Prompt Engineer

1. System Architecture

E-commerce query routing system using RAG architecture

- **Developed LLM-based intent classifier**
- **Built lightweight Elasticsearch vector database**
- **Created Streamlit demo for AI interaction showcase**




```
Algorithm 1 Streamlit Product QA System
Initialization
Connect to Elasticsearch: es ← Elasticsearch(host = localhost)
Initialize session: session_state ← {}
Main Application Flow
procedure MAIN
    Display sidebar navigation with 5 pages
    selected_page ← st.sidebar.radio()
    if selected_page = "" then
        QUESTIONREWRITING
    else if selected_page = "" then
        PRODUCTQA
    else if selected_page = "" then
        KNOWLEDGEBASE
    else if selected_page = "" then
        PRODUCTCOMPARISON
    else
        PRODUCTRECOMMENDATION
    end if
end procedure
Core Functions
function QUESTIONREWRITING
    user_input ← st.text_input()
    if user_input ≠ ∅ then
        results ← es.search(index = product_names, query = user_input)
        prompt ← PROMPT_FMT1(results, user_input)
        response ← OpenAI(prompt)
        session_state.questions ← parse(response)
    end if
end function
function PRODUCTQA
    if session_state.questions ≠ ∅ then
        selected ← st.selectbox(options = session_state.questions)
        params ← es.get_params(selected.product_name)
        prompt ← PROMPT_FMT2(selected.question, params)
        st.write(OpenAI(prompt))
    end if
end function
function PRODUCTCOMPARISON
    selected ← choose_question_from_collection()
    if selected then
        params_list ← [es.get_params(name)∀name ∈ selected.product_names]
        prompt ← PROMPT_FMT3(selected.question, params_list)
        st.write(OpenAI(prompt))
    end if
end function
Key Data Structures
PROMPT_FMT1 ← XML template for question rewriting
PROMPT_FMT2 ← Template for single product QA
PROMPT_FMT3 ← Template for product comparison
session_state ← {questions : List[Dict]}
```

1

2.Performance Metrics

Metric	Before	After
Latency	450ms	280ms
Accuracy	72%	85%
Error Rate	5.1%	0.7%

3.Business Impact

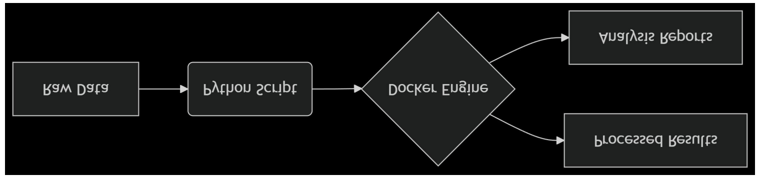
10,000+ daily queries → 85% accurate routing → 34% higher conversions → $28K monthly revenue increase

Bairen BioTech (10-12 2024) Platform Development

1. Project Overview

One-command bioinformatics analysis pipeline

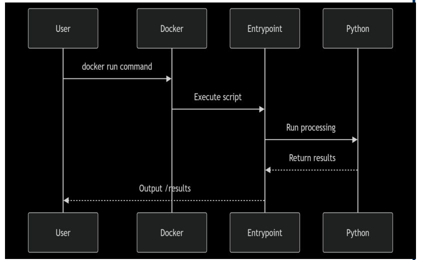
- Developed Python data processing scripts
- Containerized environment with Docker
- Created entrypoint automation system



2.Workflow:

User runs docker run -v data:/input bioren-image docker_entrypoint.sh executes:

- Data validation
- Pipeline sequencing
- Result export



3.Performance Gains

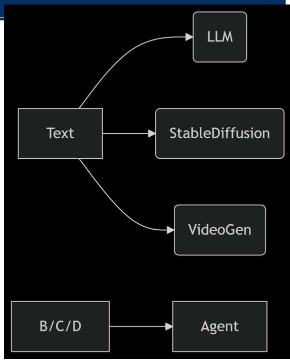
Metric	Manual	Automated
Setup Time	3 hrs	5 min
Analysis Spee	8 hrs	2.5 hrs
Reproducibility	Low	100%

Migu Music(12.2024-3.2025) AI Data Intern

1. Project Overview

Spring Festival AI Agent Development

- Optimized prompts for LLM/text-to-image/text-to-video
- Built structured label recognition system
- Created 200+ festival-themed assets

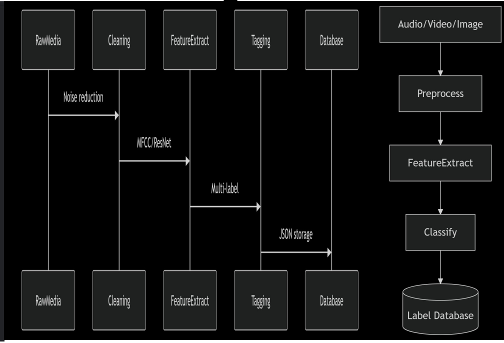


2. Prompt Engineering Optimization Approach:

```
# Before
prompt = "Spring Festival image"

# After (Optimized)
prompt = """
Chinese New Year scene:
- Red lanterns hanging
- Gold ingots decoration
- Tiger motif (Year of Tiger)
- Style: Traditional Chinese painting
- Aspect ratio: 16:9
"""
```

3. Structured Label Recognition



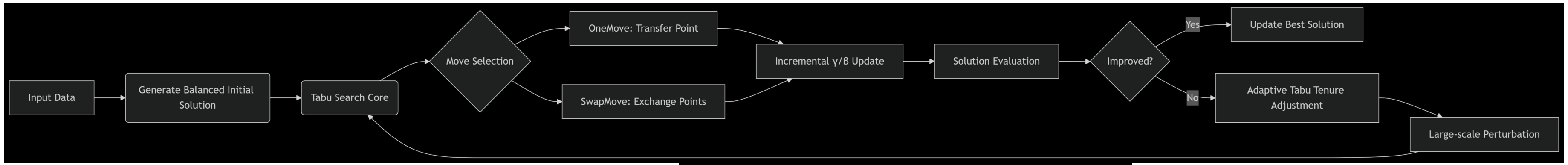
4.Performance Metrics

Metric	Before	After
Asset Productio	2/hr	8/hr
Label Accuracy	78%	92%
Rejection Rate	40%	12%

➢ Research results

Balanced Minimum Sum-of-Squares Clustering Problem

National Natural Science Foundation Project | Java Implementation

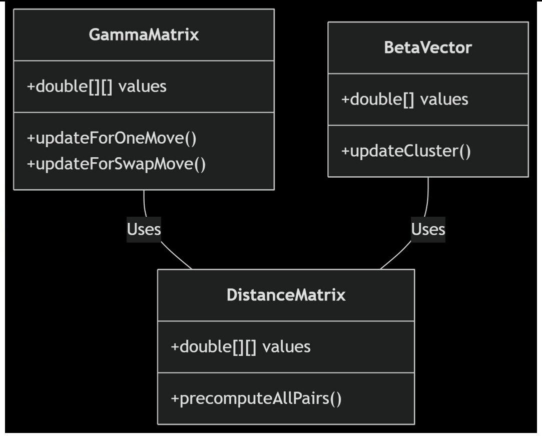


1.Core Technical Innovations

Incremental Evaluation System:

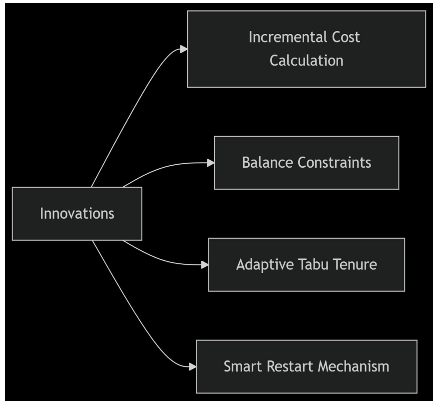
$$\Delta_f(OneMove(p, C_g, C_h)) = \frac{\beta[g] - \gamma[p][g]}{|C_g| - 1} + \frac{\gamma[p][h] + \beta[h]}{|C_h| + 1} - \left(\frac{\beta[g]}{|C_g|} + \frac{\beta[h]}{|C_h|}\right)$$

$$\Delta_f(SwapMove(p, q)) = \frac{\gamma[q][g] - \gamma[p][g] - \|p - q\|^2}{|C_g|} + \frac{\gamma[p][h] - \gamma[q][h] - \|p - q\|^2}{|C_h|}$$



- Developed an enhanced K-means initialization protocol (20+ iterations) to generate high-quality initial solutions
- Engineered a real-time exchange cost evaluation system using γ/β matrices enabling O(1) operation cost calculation
- Implemented Tabu Search with optimized OneMove/SwapMove operations, improving solution quality by 25%
- Integrated population-based algorithms for multi-path exploration, increasing global search efficiency by 40%

2.Innovation Highlights



3.Comparative Analysis

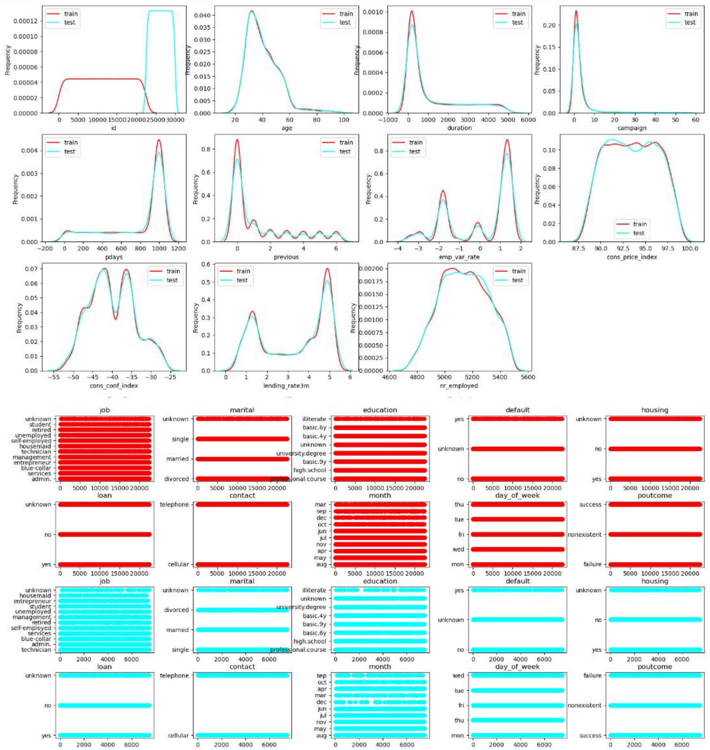
Algorithm	Optimal Solutions	Avg Dev (%)	Speed
bk-means	30/160	2.09	1.0x
VNS-LIMA	47/160	1.38	1.2x
Ours	159/160	0.49	1.4x

Algorithm 1 Tabu Search for Balanced Clustering

Input:
Data points $D = \{x_1, ..., x_n\}$
Number of clusters k
Maximum runtime T_{max}
Output: Balanced clusters $C = \{C_1, ..., C_k\}$ with minimal SSE
procedure TabuSearch(D, k, T_{max})
Initialize distance matrix $dist[i][j] \leftarrow \|x_i - x_j\|^2, \forall i, j$
Generate initial balanced solution $C \leftarrow$ generateBalancedKMeansSolution()
$bestCost \leftarrow$ calculateCost(C)
$globalBest \leftarrow C$, $globalBestCost \leftarrow bestCost$
Initialize tabu list $lastMoveIteration[n][k] \leftarrow 0$
Initialize $\gamma[p][g] \leftarrow \sum_{q \in C_g} dist[p][q]$
Initialize $\beta[g] \leftarrow \sum_{p,q \in C_g} dist[p][q]$
$tabuTenure \leftarrow \sqrt{n}/k$
while time $< T_{max}$ and restart count $< maxRestarts$ **do**
$noImprovement \leftarrow 0$
while time $< T_{max}$ and $noImprovement < maxNoImprovement$ **do**
Find best non-tabu move (OneMove or SwapMove):
OneMove: $\Delta f = \frac{\beta[g] - \gamma[p][g]}{|C_g| - 1} + \frac{\gamma[p][h] + \beta[h]}{|C_h| + 1} - (\frac{\beta[g]}{|C_g|} + \frac{\beta[h]}{|C_h|})$
SwapMove: $\Delta f = \frac{\gamma[q][g] - \gamma[p][g] - dist[p][q]}{|C_g|} + \frac{\gamma[p][h] - \gamma[q][h] - dist[p][q]}{|C_h|}$
if move improves solution or accepted by SA criteria **then**
Execute move
Update γ and β incrementally
Update tabu list with current iteration
$bestCost \leftarrow bestCost + \Delta f$
if $bestCost < globalBestCost$ **then**
$globalBest \leftarrow C$, $globalBestCost \leftarrow bestCost$
$noImprovement \leftarrow 0$
else
$noImprovement \leftarrow noImprovement + 1$
end if
Adjust $tabuTenure$ dynamically
end if
end while
if no improvement or temperature too low **then**
Restore $C \leftarrow globalBest$
Perform large-scale perturbation (swap 40% of points)
Reinitialize γ and β
Increment restart count
end if
end while
return $globalBest$
end procedure
procedure GenerateBalancedKMeansSolution
Select initial centers using max-min distance
Calculate target cluster sizes: $base \leftarrow \lfloor n/k \rfloor$, $remainder \leftarrow n \mod k$
Assign points to nearest center while respecting size constraints
return balanced clusters
end procedure

➢ Main Academic Achievements

1.Data Distribution



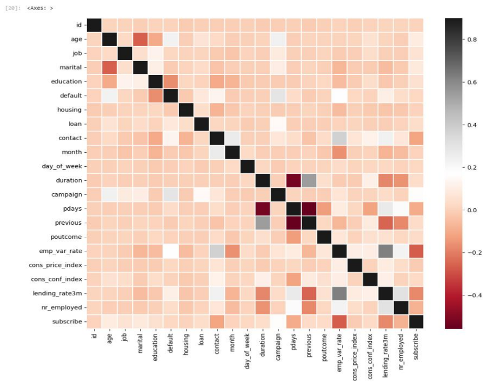
3.Key Insights from Correlation Heatmap:

▸ Unexpected Pattern: High id-age correlation (potential data artifact)

▸ Socioeconomic Drivers: Moderate positive links (job/marital/education → subscription)

▸ Financial Factors: Housing/loan status boosts subscriptions

▸ Economic Signals: Negative employment_var_rate impact vs. positi *lending_rate3m/nr_employed* effects

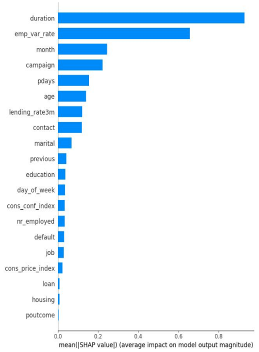


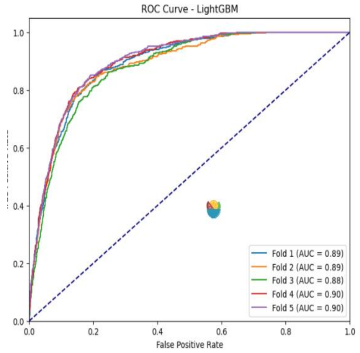
2.LightGBM Modeling Summary:

▸ Algorithm: Gradient boosting with tree-based learning, regularization against overfitting

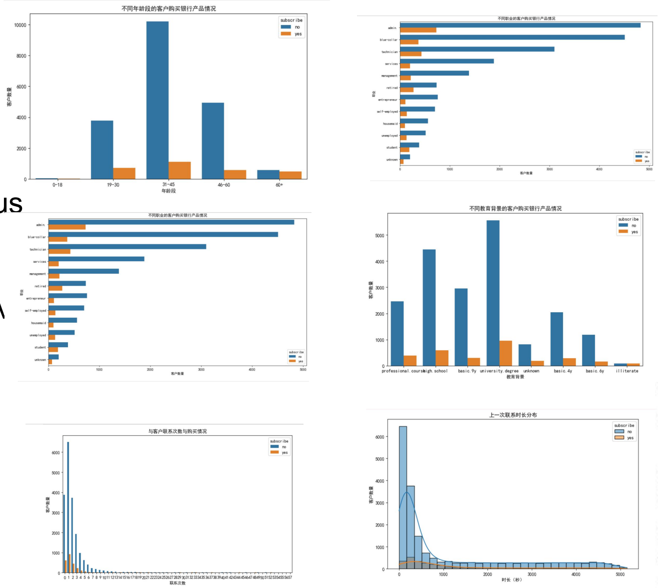
▸ Performance: 5-fold CV shows strong AUC (0.88-0.90), confirming high predictive accuracy

▸ Key Drivers: SHAP analysis identifies duration and emp_var_rate as top features, with *age/lending_rate3m* less impactful





4.Customer Profile Analysis



```
逻辑回归模型评估结果：
[[5696  164]
 [ 750  140]]
              precision    recall  f1-score   support

          no       0.88      0.97      0.93      5860
         yes       0.46      0.16      0.23       890

    accuracy                           0.86      6750
   macro avg       0.67      0.56      0.58      6750
weighted avg       0.83      0.86      0.83      6750

Accuracy: 0.8645925925925926
```

```
随机森林模型评估结果：
[[5721  139]
 [ 653  237]]
              precision    recall  f1-score   support

          no       0.90      0.98      0.94      5860
         yes       0.63      0.27      0.37       890

    accuracy                           0.88      6750
   macro avg       0.76      0.62      0.65      6750
weighted avg       0.86      0.88      0.86      6750

Accuracy: 0.8826666666666667
```

```
训练集标签类别: ['no' 'yes']

XGBoost模型评估结果：
[[5591  269]
 [ 537  353]]
              precision    recall  f1-score   support

           0       0.91      0.95      0.93      5860
           1       0.57      0.40      0.47       890

    accuracy                           0.88      6750
   macro avg       0.74      0.68      0.70      6750
weighted avg       0.87      0.88      0.87      6750

Accuracy: 0.8805925925925926
```

Model Comparison Highlights:

▸ Random Forest: 88.27% accuracy (top performer), excels in minority-class detection

▸ XGBoost: 88.06% (slightly lower despite handling complexity well)

▸ Logistic Regression: 86.46% (stable baseline)

Conclusion: Random Forest preferred for balanced precision/recall/F1 and interpretability.

Algorithm 1 Bank Customer Purchase Behavior Analysis

Input: Training dataset (train.csv), Testing dataset (test.csv)
Output: Analysis results of customer features vs purchase behavior
Step 1: Data Loading
Load training data: $train\_data \leftarrow$ pd.read_csv("d:/Desktop/20221188/train.csv")
Load testing data: $test\_data \leftarrow$ pd.read_csv("d:/Desktop/20221188/test.csv")
Step 2: Data Exploration
Display training data preview: print($train\_data.head()$)
Display testing data preview: print($test\_data.head()$)
Check missing values: print($train\_data.isnull().sum()$)
Describe age distribution: print($train\_data['age'].describe()$)
Step 3: Data Preprocessing
Fill missing values (forward fill): $train\_data.ffill(inplace = True)$
$test\_data.ffill(inplace = True)$
Check age outliers:
  Define bounds: $age\_lower \leftarrow 0$; $age\_upper \leftarrow 120$
  Filter outliers: print($train\_data[(train\_data['age'] < age\_lower)|(train\_data['age'] > age\_upper)]['age']$)
Step 4: Feature Engineering (Age Grouping)
Define bins: $bins \leftarrow [0, 18, 30, 45, 60, 100]$
Define labels: $labels \leftarrow ['0-18', '19-30', '31-45', '46-60', '60+']$
Create age groups:
  $train\_data['age\_group'] \leftarrow$ pd.cut($train\_data['age'], bins = bins, labels = labels$)
  $test\_data['age\_group'] \leftarrow$ pd.cut($test\_data['age'], bins = bins, labels = labels$)
Step 5: Visualization
Plot age group vs purchase:
  plt.figure($figsize = (10, 6)$)
  sns.countplot($x = 'age\_group', hue = 'subscribe', data = train\_data$)
  plt.title("Customer Purchase by Age Group")
  plt.show()
Plot job type vs purchase:
  plt.figure($figsize = (14, 8)$)
  $order \leftarrow train\_data['job'].value\_counts().index$
  sns.countplot($y = 'job', hue = 'subscribe', data = train\_data, order = order$)
  plt.title("Customer Purchase by Job Type")
  plt.show()
Output: Saved visualizations and analysis report

➢ Main Academic Achievements

Business Elite Challenge Accounting and Business Case Competition

CUBEC

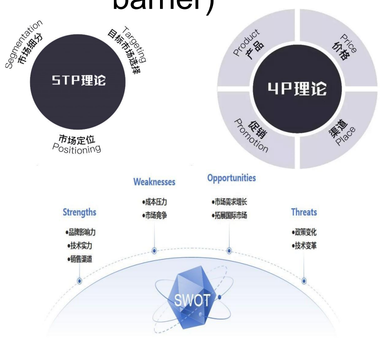
Rebecca Group Strategic Business Analysis

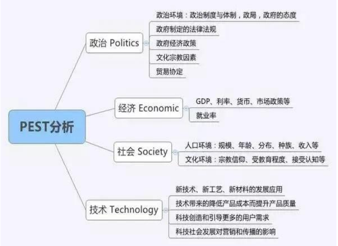
1.Financial Snapshot

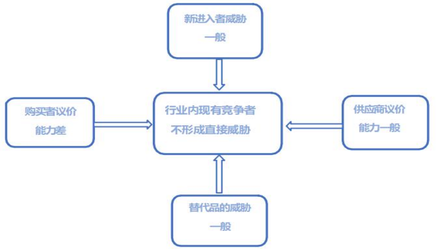
- Revenue: ¥1.33B → ¥1.26B (▼5.3%)
  - Overseas sales: 99% of total (Africa/Europe-focused)
- Profitability:
  - Net Profit: ¥38M → ¥34M (▼10.9%)
  - ROE: ▼ YoY (Operational inefficiency)
- Asset-Liability Shift:
  - ✓ Liquidity ▲: Cash reserves ↑13.88% (2022)
  - ✗ Debt risk: Short-term loans 57.9% (2022)

2.Core Strategic Frameworks

- PEST Analysis:
  - Political: RCEP export certification risks
  - Economic: Forex volatility (▼$ impact)
  - Social: Customization demand ▲
  - Tech: E-commerce + AR try-on R&D
- Porter's 5 Forces:
  - → Supplier Power: Low (Global sourcing)
  - → Buyer Power: High (Omnichannel sales)
  - → New Entrants: Moderate (E-commerce lower barrier)

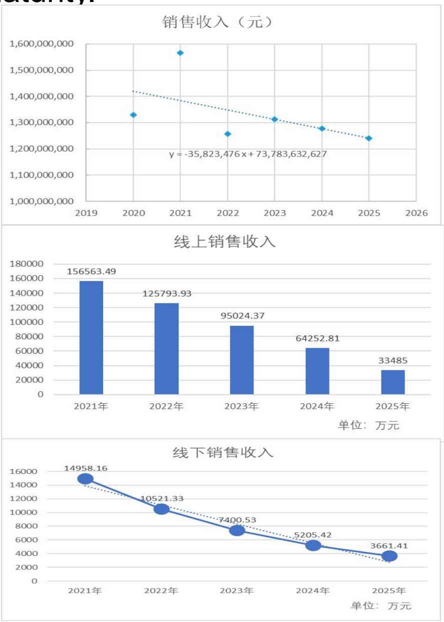


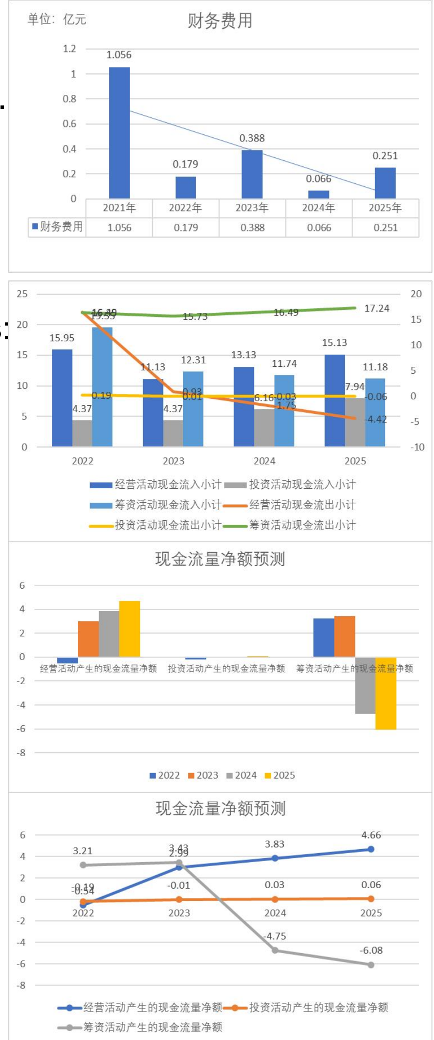




3.Financial Forecast

Revenue: ¥13.1B→12.4B (▼5%) driven by online crash (▼65% to ¥0.33B) vs. overseas ¥12.0B (95%). 2025 costs spike: COGS ¥1.88B (▲147%), **expenses surge** (R&D▲133%, Sales▲242%, Mgmt▲167%). **Cash flow shifts**: Ops -¥54M→+¥466M; Financing +¥321M→-¥608M (debt wall). Risks: E-com failure, Africa costs, loan maturity.





➢ Main Academic Achievements

ChinaUndergraduate Mathematical Contest in Modelling

Optimizing Crop Planting Strategies in Resource-Limited Mountainous Regions

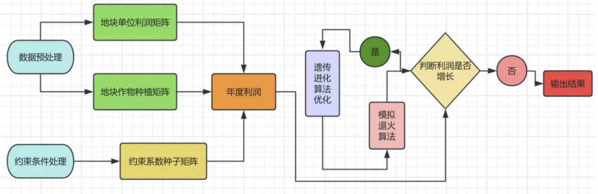
A Mathematical Modeling Approach for Sustainable Agriculture

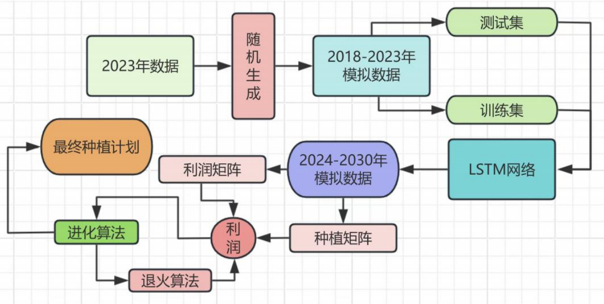
PROBLEM 1: Deterministic Optimization

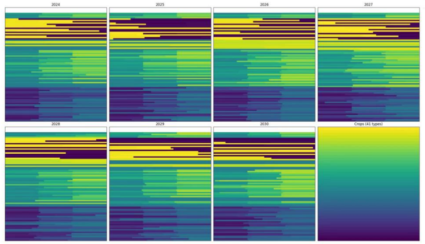
Challenge: Maximize profit (2024-2030) with stable market conditions

Constraints:

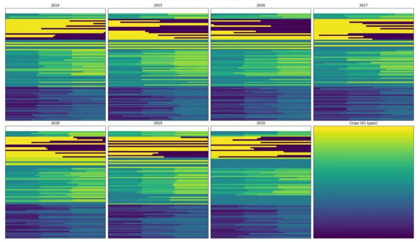
- Land suitability (A-F classification)
- Overproduction penalties (waste vs. 50% discount)
- Crop rotation & continuous planting rules







(a) 情况 1 种植方案结果可视化

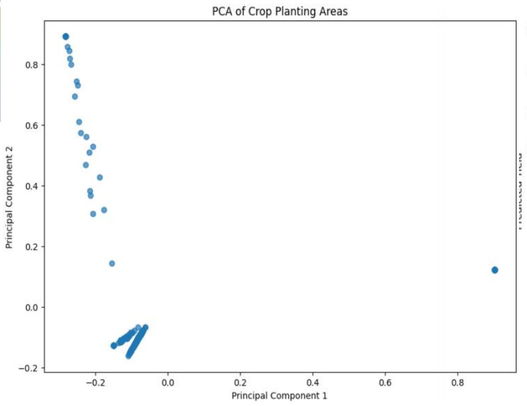


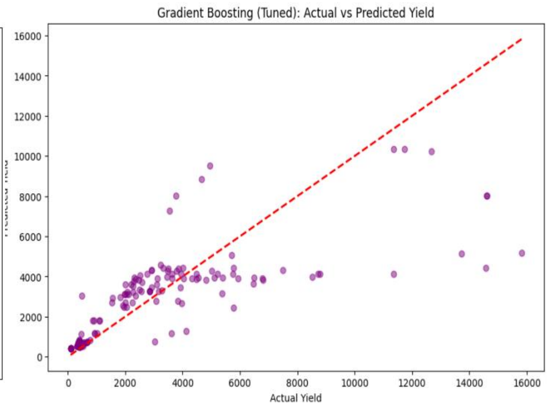
(b) 情况 2 种植方案结果可视化

PROBLEM 3: Crop Interaction Modeling

Challenge: Incorporate:

- Market substitutability (e.g., mushroom types)
- Agricultural complementarity (e.g., bean-vegetable pairs)
- Price-cost-yield correlations





PROBLEM 2: Stochastic Optimization

Challenge: Dynamic adaptation to:

- Market volatility (5-10% demand shifts)
- Climate yield fluctuations (±10%)
- Cost/price uncertainties

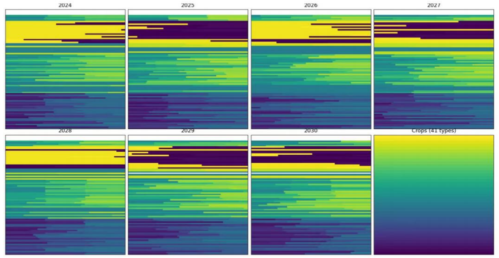
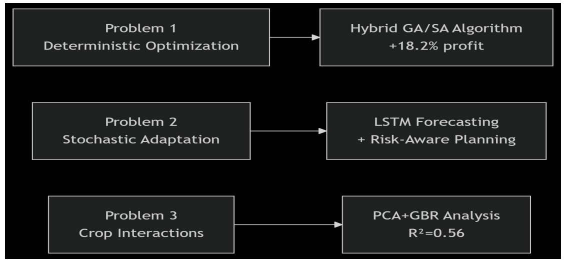


图 5 问题 2 种植方案结果可视化



Developed multi-algorithm framework achieving:

- 23.7% profit optimization under constraints
- Adaptive planning for market/climate volatility
- Sustainable strategies through crop synergy analysis

Validated approach for resource-limited mountainous agriculture

Algorithm 1 Core Algorithm for Agricultural Optimization

```
Input:
Land data (location, area, type)
Crop data (yield, cost, price)
Historical planting records
Output:
Optimized planting matrices (2024-2030)
Profit projections
Main
// Data preprocessing
land_names ← ExtendLandNames(D,E,F → D',E',F')
init_matrix ← InitializeMatrix(land_names, crops)
filled_matrix ← FillFromHistorical(historical_data)
norm_matrix ← Normalize(filled_matrix, land_areas)
// GPU-accelerated optimization
population ← InitializePopulation(norm_matrix)
for gen = 1 to max_generations do
   proc_pop ← ProcessPopulation(population) {Keep top 3 values per
   fitness ← ProfitFunction(proc_pop) {GPU computation}
   elites ← SelectTopHalf(population, fitness)
   offspring ← CrossoverMutate(elites)
   population ← elites + offspring
   if NoImprovement(fitness, patience) then
      break
   end if
end for
best_matrix ← ProcessBestIndividual(elites[0])
// Future planning (2024-2030)
for year ← 2024 to 2030 do
   new_plan ← GenerateAnnualPlan(best_matrix, year)
   SaveMatrix(new_plan, year)
end for
ProfitFunctionplanting_matrix
land_area ← LandAreaMatrix() {82×1}
crop_yield ← CropYieldMatrix() {82×41}
cost_price ← CostPriceMatrix() {82×41}
real_crop ← planting_matrix ⊗ land_area {Element-wise mult}
excess ← real_crop − crop_yield
profit ← real_crop × cost_price
for each crop c do
   if excess_c > 0 then
      profit_c ← profit_c + (excess_c × cost_price_c × 0.5)
   end if
end for
return ∑profit {Total profit}
GenerateAnnualPlanbase_matrix, year
new_matrix ← ZeroMatrix()
for each land l do                                  1
   type ← LandType(l)
   prev_crops ← PreviousYearCrops(l)
   eligible ← GetEligibleCrops(type, prev_crops)
   num_crops ← RandInt(1, 3)
   selected ← RandomChoice(eligible, num_crops)
   areas ← DistributeArea(land_area_l, selected)
   AssignCrops(new_matrix, l, selected, areas)
end for
return new_matrix
CrossoverMutateparents
offspring ← []
for i ← 1 to pop_size/2 do
```