

Team Number :	apmcm24205675
Problem Chosen :	C

2024 APMCM summary sheet

With the evolution of consumption patterns and rising living standards, the pet industry has emerged as a significant economic sector globally. This study analyzes China's pet industry development (2019-2023), focusing on the pet food sector's market dynamics, industry structure, and policy impacts to provide evidence-based recommendations.

Our initial analysis of three datasets revealed significant market trends: **China's cat population showed robust growth** (12.15% annually, CV=18.97%) **while dogs declined** (-1.52%, CV=3.16%). International markets displayed contrasting patterns, with U.S. declining despite market dominance and European markets showing stable growth. Data preprocessing confirmed high dataset quality for subsequent modeling.

In addressing the first question regarding China's pet industry development, Initially, we collected and preprocessed additional data .Through feature extraction and statistical analysis, we identified distinct trends: the cat population showed robust growthwhile the dog population experienced a slight decline.Using **correlation analysis and multiple regression models**, we identified key influencing factors: the cat market was primarily driven by market size (691.52), GDP per capita (191.20), and population growth (105.55), while the dog market was more influenced by population growth (1067.06), market size (669.89), and GDP per capita (326.20).For future predictions, we employed multiple models including SARIMA, Holt-Winters, and BayesianRidge, with the latter proving most effective.The BayesianRidge model, validated through **parameter sensitivity analysis and bootstrap interval estimation**, **projected continued growth in the cat population while suggesting stability in the dog population for the next three years, with notably higher prediction reliability for the dog market segment.**

In analyzing global pet food demand, we conducted a comprehensive examination of the global pet industry through multi-dimensional analysis.**The market exhibits a distinct "dual-core+multiple" pattern**, with the US dominating (64.06% market share) and China emerging strongly (26.99% share, 8.20% growth rate).Our analysis revealed significant structural differences across markets: while the US maintains a balanced cat-dog ratio (48%-52%), other markets show a clear "cat dominance" trend, particularly evident in France (62.6% cats). For demand forecasting, we developed nine specialized features and applied **VIF filtering** to address collinearity.Using multiple regression models, with Linear Regression emerging as the most effective ( $R^2=0.99$ ), we projected pet food demand for 2024-2026.Sensitivity analysis identified three critical factors: total pet expenditure (elasticity coefficient 0.55), cat-to-dog ratio (0.53), and pet GDP per capita (0.34).The model predicts **sustained growth across all markets, with China showing the most dynamic expansion potential, while mature markets like the US demonstrate stable growth patterns.**

In analyzing China's pet food industry development, we introduced three key indicators for trend analysis: **Industry Scale Growth Rate, Export Dependency, and Output Efficiency.**The analysis revealed significant industry characteristics: a robust average annual growth rate of 65.2% with high volatility (49.6%), indicating rapid but unstable development; declining export dependency from 34.9% to 9.9%, suggesting stronger domestic market orientation;and remarkable output efficiency growth of 385.8%,reflecting technological and management improvements.For future projections, due to limited data points and high feature collinearity, we employed simplified regression models focusing on production volume, export volume, and market size.The BayesianRidge model predicted steady growth in both production (from 3537.59 in 2024 to 5945.06 in 2026) and exports (from 38.00 to 55.62), with relatively narrow confidence intervals.These findings indicate that **China's pet food industry is transitioning from export-oriented to domestic market-driven, with significant potential for further growth through technological advancement and market expansion.**

In analyzing policy impacts and developing strategic recommendations for China's pet food industry, we established comprehensive quantitative indicators including **tariff elasticity, market growth rates, and policy sensitivity**

**measures.** The analysis revealed a distinct **"domestic-sales-dominated" structure** (98.85% domestic sales ratio) with concentrated export patterns (US market share 85.41%, HHI=7402). The impact assessment showed varying effects across dimensions: high cost impact from tariff changes (73.33% reduction), low market and structural impacts (export ratio change -0.07%), and medium risk impact. Based on these findings, **we proposed a multi-faceted development strategy emphasizing "domestic demand leadership, innovation drive, structural optimization, and risk control."** The strategy recommends differentiated approaches for cat and dog markets, market diversification to reduce US market dependency, and a three-step industrial upgrading plan. Special attention is given to the growing "cat-oriented" trend and the need for flexible policy response mechanisms, despite moderate tariff sensitivity, to ensure sustainable industry development.

The primary strengths of our methodology lie in its **comprehensive multi-dimensional indicator framework** and **the adoption of robust interval estimation techniques** to overcome the limitations of classical statistical inference in small sample scenarios.

***Keywords: multi-model predictive analytics ; comprehensive indicator assessment ; robust interval estimation ; structural market analysis ;***

## Contents

<b>1 Introduction .....</b>	<b>1</b>
1.1 Background .....	1
1.2 Work .....	1
<b>2 Problem analysis .....</b>	<b>2</b>
2.1 Analysis of Question I .....	2
2.2 Analysis of Question II .....	2
2.3 Analysis of Question III .....	2
2.4 Analysis of Question IV .....	2
<b>3 Symbol and Assumptions .....</b>	<b>4</b>
3.1 Symbol Description .....	4
3.2 Fundamental assumptions .....	4
<b>4 Descriptive statistics and preprocessing of attachment data .....</b>	<b>5</b>
4.1 Descriptive statistics .....	5
4.2 Data preprocessing .....	8
<b>5 Model and test .....</b>	<b>9</b>
5.1 Question I: Analysis of the development of pet industry in China .....	9
5.1.1 Additional data collection .....	9
5.1.2 Analyze the development of China's pet industry in the past 5 years .....	10
5.1.3 Analyze the factors affecting the development of China's pet industry .....	12
5.1.4 Forecast the development trend in the next three years .....	15
5.1.5 Parameter sensitivity analysis of BayesianRidge .....	17
5.2 Question II: Global pet food demand forecast for the next three years .....	18
5.2.1 Global pet industry development analysis .....	18
5.2.2 Pet food spending forecast .....	20
5.2.3 Sensitivity analysis of linear regression model .....	18
5.3 Question III: Analysis of pet food industry in China .....	23
5.3.1 Food industry development trend analysis .....	23
5.3.2 Forecast of import and export volume of pet food industry in China .....	24
5.4 Question IV: Policy quantitative analysis and development strategy designation of China's pet food industry .....	26
5.4.1 Policy quantitative analysis of pet food industry in China	
5.4.2 Put forward the development strategy	
<b>6 Strengths and Weakness .....</b>	<b>29</b>
6.1 Strengths .....	29
6.2 Weakness .....	30
<b>References .....</b>	<b>30</b>
<b>Appendix .....</b>	<b>31</b>

# 1 Introduction

## 1.1 Background

With the change of people's consumption concept, the pet industry, as an emerging industry, has gradually emerged around the world, which benefits from the rapid development of the economy and the improvement of per capita income. In 1992, the China Small Animal Protection Association was established, followed by international pet brands such as Royal Canin and Mars entering the Chinese market in 1993. Pet-related industries, such as pet food, pet clinics, pet supplies and pet care, have also gradually gained a broad and rapid growth space in the Chinese market, because "pet companionship" is becoming more and more popular in China. Based on the attached data and additional data collected by the team, please analyze the development trend and market demand of the pet industry. Based on the analysis results and the current economic environment, please put forward corresponding strategic suggestions for the development of China's pet industry.

## 1.2 Work

- Problem 1:

Based on Attachment 1 and additional data collected by the team, please analyze the development of China's pet industry by pet type in the past five years. And analyze the factors affecting the development of China's pet industry, so as to establish an appropriate mathematical model to predict the development of China's pet industry in the next three years.

- Problem 2:

The overseas pet industry, such as European countries and the United States, has also developed rapidly in recent years. Please analyze the development of the global pet industry by pet type based on Attachment 2 and additional data collected by the team. And build appropriate mathematical models to predict the global demand for pet food in the next three years.

- Problem 3:

According to the production and export value of pet food in China in Annex 3, please analyze the development situation of pet food industry in China and forecast the production and export situation of pet food in China in the next three years (regardless of economic policy changes), based on the global pet food market demand and the development trend in China.

- Problem 4:

China's pet food industry will inevitably be affected by the new economic policies (such as tariffs) of European countries and the United States. To quantitatively analyze this effect, build appropriate mathematical models and take into account the attached data, the additional data collected by the team, and the calculations for the above questions. Based on the calculation results, please formulate feasible strategies for the sustainable development of China's pet food industry.

## **2 Problem analysis**

### **2.1 Analysis of Question I**

In order to analyze the development of China's pet industry, we can first establish an analysis basis through descriptive statistics and data preprocessing, focusing on statistical indicators such as mean value, standard deviation and coefficient of variation. On this basis, through the analysis of the changing trend of the number of cats and dogs and the evolution of market structure, the development characteristics of the industry are revealed. Further, through correlation analysis and multiple regression modeling, the key influencing factors were identified and quantified, and their importance was determined. Finally, several prediction models such as SARIMA, Holt-Winters and BayesianRidge were compared, combined with parameter sensitivity analysis and interval estimation, to forecast the future development trend of the industry.

### **2.2 Analysis of Question II**

In order to forecast the global pet food demand, we can first analyze the structural characteristics of the global market and compare the development characteristics of different regional markets. Through in-depth analysis of the proportion structure and maturity differences of cats and dogs in each market, the development stage characteristics of the market are revealed. In the prediction modeling stage, by constructing multiple feature indexes, VIF method is used to deal with the feature collinearity problem, and the prediction model is established. Finally, through sensitivity analysis, key influencing factors were identified and their elastic coefficients were calculated to provide reliable support for the prediction results.

### **2.3 Analysis of Question III**

When analyzing the development of China's pet food industry, we can design three core indicators of industrial scale growth rate, export dependence and output efficiency for trend analysis. These indicators reveal the development characteristics and trends of the industry. Considering the limitations of limited data points and high feature collinearity, a simplified regression model can be used to forecast the output, export volume and market size, and the development trend of the industry can be verified by the prediction results of the model.

### **2.4 Analysis of Question IV**

In order to conduct quantitative policy analysis and formulate development strategies, we can establish a comprehensive quantitative index system including tariff flexibility, market growth rate and policy sensitivity. Through the analysis, the structural characteristics of the industry are found, and the risk of export market concentration is measured by the HHI index. Based on the results of the multidimensional impact assessment, the development strategy is proposed, and specific suggestions are put forward for the differentiation characteristics of the cat and dog market, the diversified demand of the export market and the path of industrial upgrading.

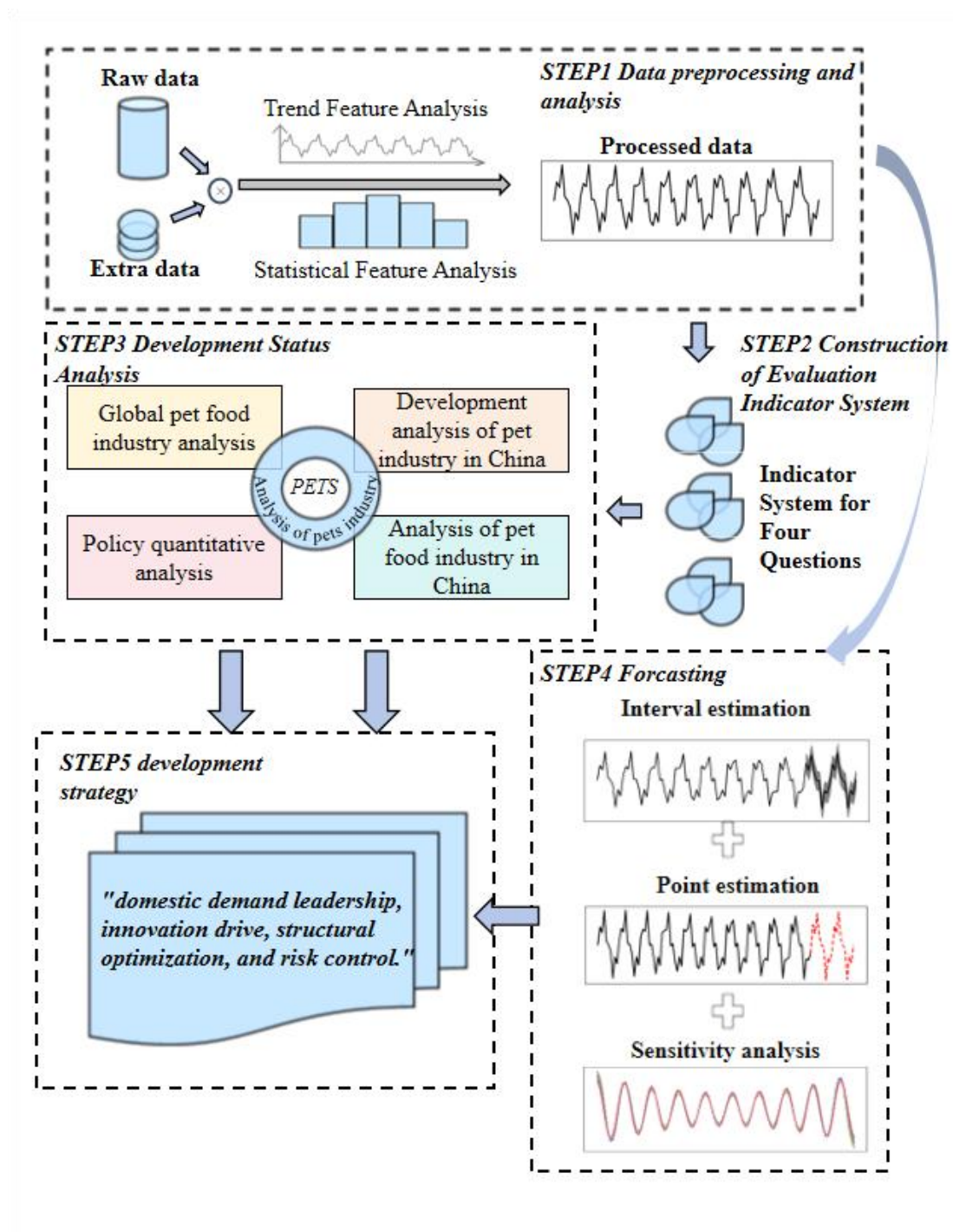


Figure 0 General idea diagram

### 3 Symbol and Assumptions

#### 3.1 Symbol Description

<i>Symbol</i>	<i>Description</i>
$CV$	<i>Coefficient of Variation</i>
$y_i$	<i>dependent variable</i>
$\epsilon_i$	<i>Error term</i>
$HHI$	<i>Market Concentration</i>
$r$	<i>Correlation Coefficient</i>
$Cat\ Growth_t$	<i>The annual growth rate of the cat population</i>
$Total\ Pets_t$	<i>Total pet population</i>
$Cat\ MA_t$	<i>2 year rolling average of cat population</i>
$Dog\ MA_t$	<i>2 year rolling average of dog population</i>

#### 3.2 Fundamental assumptions

Assumption 1 : Parameter stability: It is assumed that the model parameters are stable over different time periods and do not change significantly.

Assumption 2 : External factors: Assume that the effects of external factors (such as policy changes, market fluctuations, etc.) can be reasonably incorporated into the model.

Assumption 3 : Random error: Assume that the random error in the model is unpredictable and has a mean of zero.

## 4 Descriptive statistics and preprocessing of attachment data

### 4.1 Descriptive statistics

First, we conducted detailed descriptive statistics on the data to gain a deeper understanding of the data. In addition to common descriptive statistics such as mean value, standard deviation, maximum minimum value and median, we also introduced the following descriptive statistics for further analysis

#### (1) Coefficient of Variation (CV)

The coefficient of variation is the ratio of the standard deviation to the mean, usually expressed as a percentage. It is used to compare the dispersion of different datasets, especially when the means of the datasets are different.

$$CV = \left( \frac{\sigma}{\mu} \right) \times 100\%$$

where  $\sigma$  is the standard deviation, and  $\mu$  is the mean.

#### (2) Annual Growth Rate

The annual growth rate represents the average annual growth rate of the data over a certain period. It is used to measure the average rate of change of the data over time.

$$\text{Annual Growth Rate} = \left( \frac{V_{\text{end}}}{V_{\text{start}}} \right)^{\frac{1}{n}} - 1$$

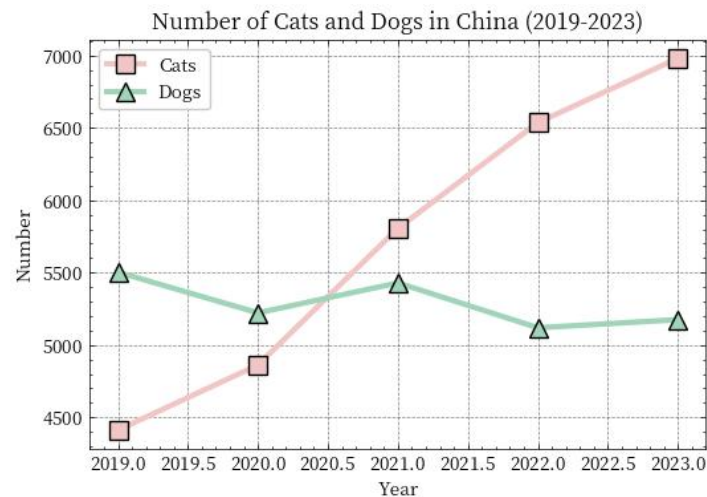
where  $V_{\text{end}}$  is the value at the end,  $V_{\text{start}}$  is the value at the start, and  $n$  is the number of years.

Attachment 1 descriptive statistical results are as follows:

**Table 1 Attachment 1 Descriptive statistical results**

Statistical index	Cat	Dog
Mean	5719.2	5289.6
Standard deviation	1084.882113	167.1669824
Minimum	4412	5119
Maximum	6980	5503
Median	5806	5222
Coefficient of variation	18.96912354	3.160295342
Annual growth rate	12.15147241	-1.524609802





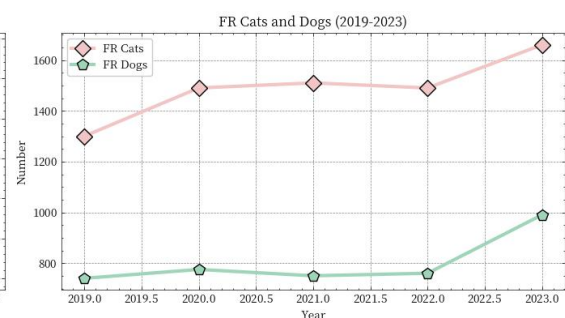
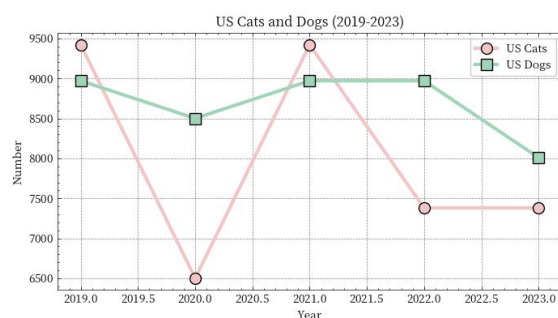
**Figure 1 Attachment 1 visualization**

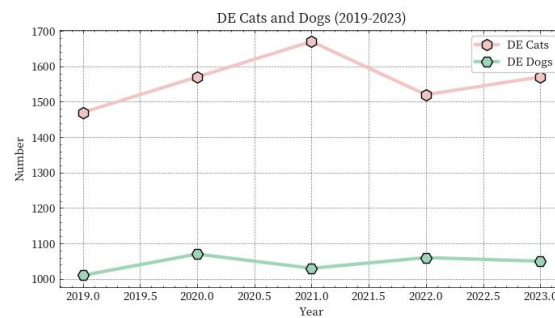
Through a detailed analysis of the statistical results in Attachment 1, we can draw the following conclusions: During 2019-2023, the number of cats presents a significant growth trend, with an average annual growth rate of 12.15% and high volatility, with a standard deviation of 1084.88 and a coefficient of variation of 18.97%, indicating that the market demand for cats is increasing. In contrast, the number of dogs showed a slight downward trend, with an average annual growth rate of -1.52%, a standard deviation of only 167.17, and a coefficient of variation of 3.16%, indicating that the market demand for dogs was relatively stable or slightly declining. The average number of cats (5719.2) and maximum (6980) were both higher than the average number (5289.6) and maximum (5503) of dogs, further supporting the conclusion that cats may be more popular than dogs. These analysis results show that **the market dynamics of cats are highly variable and may be affected by a variety of factors such as market trends and policy changes, while the market for dogs is relatively stable.**

Attachment 2 descriptive statistical results are as follows:

**Table 2 Attachment 2 Descriptive statistical results**

	Cat _ Mean	Cat _ Standa rd deviati on	Cat _ Coeffici ent of variatio n	Cat _ Averag e annual growth rate	Dog _ Mean	Dog _ Standa rd deviati on	Dog _ Coeffici ent of variatio n	Dog _ Averag e annual growth rate
<b>US</b>	8020	1327.5	16.5	-5.9	8684	428.2	4.9	-2.7
<b>FR</b>	1490	127.8	8.5	6.3	803	105.3	13.1	7.5
<b>DE</b>	1560	74.1	4.7	1.6	1044	24.0	2.3	0.9





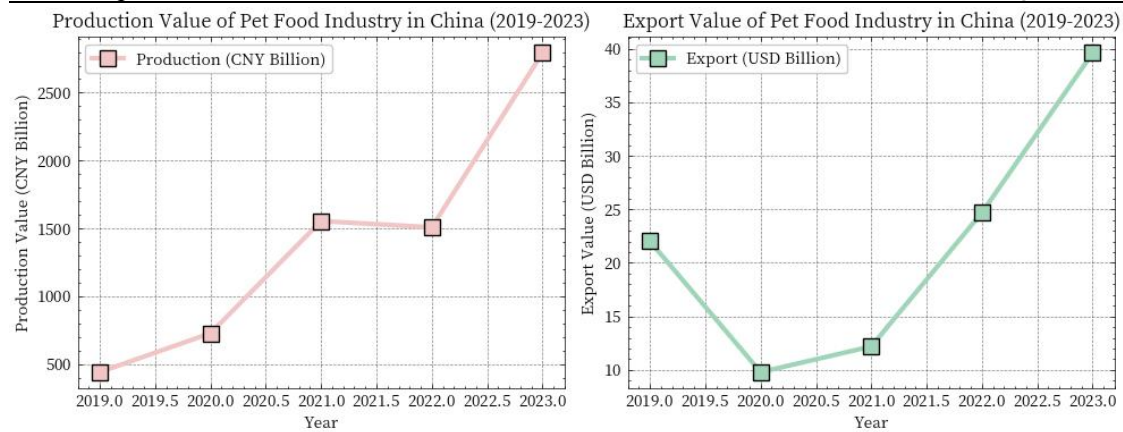
**Figure 2 Attachment 2 visualization**

Through detailed analysis of the statistical results in Attachment 2, we can draw the following conclusions: in the United States, the average number of cats is 8020, the standard deviation is 1327.5, the coefficient of variation is 16.5%, and the average annual growth rate is -5.9%, indicating that the number of cats presents a declining trend and is highly volatile. In contrast, the average number of dogs is 8684, the standard deviation is 428.2, the coefficient of variation is 4.9%, and the average annual growth rate is -2.7%, indicating that the number of dogs is also showing a downward trend, but with less volatility. In France, the average cat population is 1490, the standard deviation is 127.8, the coefficient of variation is 8.5%, and the annual growth rate is 6.3%, indicating a trend of growth and moderate volatility. The average number of dogs is 803, the standard deviation is 105.3, the coefficient of variation is 13.1%, and the average annual growth rate is 7.5%, indicating that the number of dogs also presents a significant growth trend and is highly volatile. In Germany, the average cat population is 1560, the standard deviation is 74.1, the coefficient of variation is 4.7%, and the annual growth rate is 1.6%, indicating that the cat population shows a slight trend of growth and low volatility. The mean number of dogs is 1044, the standard deviation is 24.0, the coefficient of variation is 2.3%, and the annual growth rate is 0.9%, indicating that the number of dogs also shows a slight increase trend, and the volatility is very small. **The results of these analyses show significant differences in pet market dynamics across countries, with pet populations declining in the United States and growing in France and Germany.**

Attachment 3 descriptive statistical results are as follows:

**Table 3 Attachment 3 Descriptive statistical results**

Statistical index	Gross product	Total export value
Mean	1404.6	21.66285714
Standard deviation	914.9460339	11.84653829
Minimum	440.7	9.8
Maximum	2793	39.6
Median	1508	22.01428571
Coefficient of variation	65.13925914	54.68594567
Annual growth rate	58.66530515	15.81042271



**Figure 3 Attachment 3 visualization**

The analysis results show that **both the gross product and the gross export value of China's pet food industry show a significant growth trend during 2019-2023, and the overall market performance is strong despite the high volatility of the gross product.**

Through a detailed analysis of the data, we can draw the following conclusions:

(1) In China's pet market, the number of cats shows a significant growth trend, while the number of dogs is relatively stable or slightly declining, and cats may be more popular than dogs.

(2) Among overseas markets, the US is the largest but most volatile, France and Germany are more stable and growing steadily, and cats are likely to be more popular than dogs globally.

(3) The gross product and export value of China's pet food industry have shown a significant growth trend, indicating that the market size and international competitiveness of the industry are constantly increasing.

## 4.2 Data preprocessing

In the process of data analysis and machine learning, data preprocessing is a crucial step. Data preprocessing includes processing missing values, duplicate values, and outliers to ensure data accuracy and consistency. Here is a detailed explanation of these issues:

### (1) Missing value

Missing values are situations where certain data points or eigenvalues are missing from an index data set. Missing values may be due to errors during data collection, data entry errors, or other reasons.

### (2) Duplicate value

Duplicate values are records that are identical or partially identical in the data set. Duplicate values can be due to data entry errors, data merge errors, or other reasons.

### (3) Handle outliers

Outliers are values that deviate significantly from other data points in the index data set. Outliers can be due to data entry errors, measurement errors, or other causes.

We carried out the above checks on the data of the three attachments, and used Zscores to eliminate outliers. **The results showed that the data quality of the three data sets was good, which laid a foundation for our subsequent modeling.**

## 5 Model and test

### 5.1 Question I: Analysis of the development of pet industry in China

#### 5.1.1 Additional data collection

In order to better solve problem 1, we collected additional data as follows:

**Table 4 Problem 1 Collect additional data**

Total output value of pet food	Number of pets	Pet market size	Total female population	Population growth	Per capita GDP
440.7	99.8	33.2	688179990	0.35474089	10143.86022
727.3	108.5	35.6	690171848	0.23804087	10408.71955
1554	115.4	38.9	691219627	0.0892522	12617.5051
1508	122.6	42.1	691528501	-0.013099501	12662.58317
2793	130.2	45.5	691168325	-0.103794532	12614.06099

#### (1)Total output value of pet food

This feature represents the market size of the pet food industry and is an important indicator of economic activity in the pet food industry.

Impact: An increase in total output may indicate an increase in market demand or an increase in production efficiency.

#### (2)Number of pets Number of pets

This feature indicates the size of the pet market and is an important indicator of the potential demand in the pet market.

Impact: An increase in the number of pets could indicate an expanding pet market, driving demand for pet food and other related products.

#### (3)Pet market size

This feature represents the overall size of the pet market and is an important indicator of economic activity in the pet market.

Impact: The increase in market size may indicate an increased demand for pet-related products, thus driving the growth of the pet food industry.

#### (4)Total female population

This feature indicates the potential consumer size of the target market, with women generally being the primary caregivers and consumers of pets.

Impact: An increase in the female population could indicate an increase in potential consumers in the pet market, thus boosting the pet food industry.

#### (5)Population growth

This feature indicates the rate of population growth, which affects the potential growth of the pet market.

Impact: An increase in the rate of population growth could indicate an increase in the number of potential consumers, driving the growth of the pet food industry.

#### (6)Per capita GDP

This feature indicates the purchasing power of consumers and is an important indicator to measure consumers' spending power.

Impact: An increase in GDP per capita could indicate an increase in consumer purchasing power, thus boosting the pet food industry.

**We also carried out detailed preprocessing of additional data collected in Section 4 to ensure data quality.**

### 5.1.2 Analyze the development of China's pet industry in the past 5 years

First, we perform additional feature extraction on the existing data,

#### (1) Annual growth rate

- Cat Growth: The annual growth rate of the cat population

$$\text{Cat Growth}_t = \left( \frac{\text{Cat}_t - \text{Cat}_{t-1}}{\text{Cat}_{t-1}} \right) \times 100$$

- Dog Growth: The annual increase in the number of dogs

$$\text{Dog Growth}_t = \left( \frac{\text{Dog}_t - \text{Dog}_{t-1}}{\text{Dog}_{t-1}} \right) \times 100$$

- Production Growth: The annual growth rate of production

$$\text{Production Growth}_t = \left( \frac{\text{Production}_t - \text{Production}_{t-1}}{\text{Production}_{t-1}} \right) \times 100$$

- Market Size Growth: Annual growth rate of market size

$$\text{Market\_Size\_Growth}_t = \left( \frac{\text{Market\_Size}_t - \text{Market\_Size}_{t-1}}{\text{Market\_Size}_{t-1}} \right) \times 100$$

#### (2).Market structure characteristics

- Total Pets: Total pet population

$$\text{Total Pets}_t = \text{Cat}_t + \text{Dog}_t$$

- Cat\_Ratio: Number of cats as a percentage of total pet population

$$\text{Cat Ratio}_t = \left( \frac{\text{Cat}_t}{\text{Total Pets}_t} \right) \times 100$$

- Dog Ratio: Number of dogs as a percentage of total pet population

$$\text{DoG Ratio}_t = \left( \frac{\text{Dog}_t}{\text{Total Pets}_t} \right) \times 100$$

#### (3).Per capita indicator

- GDP per capita:

$$\text{GDP per capita}_t = \frac{\text{GDP}_t}{\text{Population}_t / 10^8}$$

- Pets per capita:

$$\text{Pets per capita}_t = \frac{\text{Total Pets}_t}{\text{Population}_t / 10^8}$$

## (4)Market efficiency index

- Production\_per\_pet: Production per pet

$$\text{Production per pet}_t = \frac{\text{Production}_t}{\text{Total Pets}_t}$$

- Market Size per pet: The market size for each pet

$$\text{Market\_Size\_per\_pet}_t = \frac{\text{Market\_Size}_t}{\text{Total\_Pets}_t}$$

## (5)Paint movement average

- Cat MA: 2 year rolling average of cat population

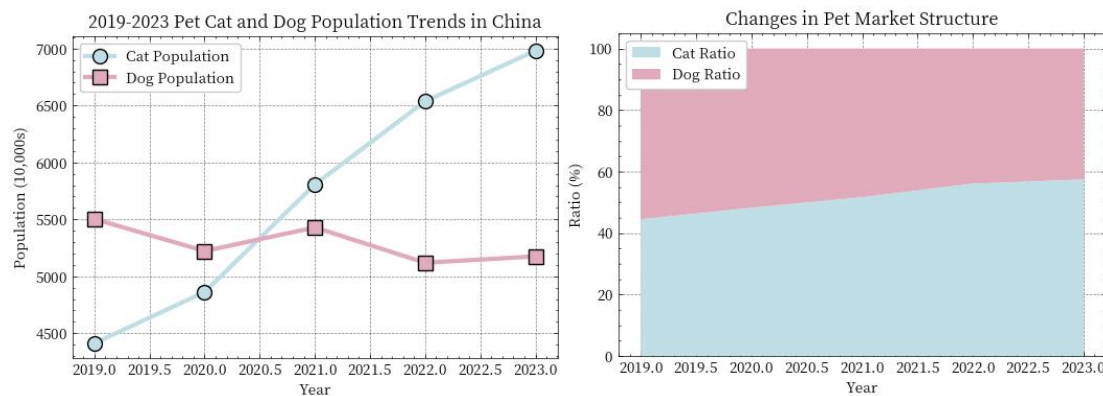
$$\text{Cat MA}_t = \frac{\text{Cat}_t + \text{Cat}_{t-1}}{2}$$

- Dog MA: 2 year rolling average of dog population

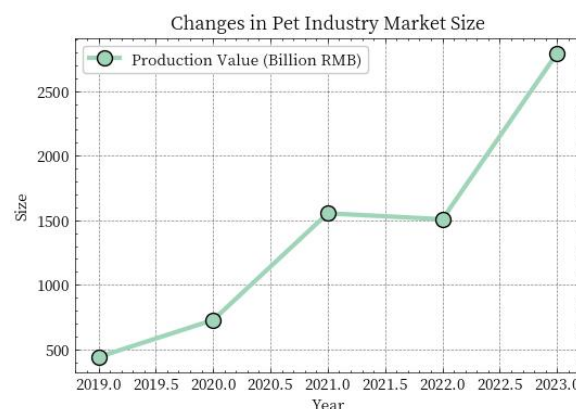
$$\text{Dog MA}_t = \frac{\text{Dog}_t + \text{Dog}_{t-1}}{2}$$

## (6).Market Development Index (Comprehensive index)

Next, we use the calculated indicators to analyze the pet development market:



**Figure 4 Cat and dog population trend and market share chart**



**Figure 5 Pet market size change chart**

We next present a detailed analysis of the above graphs:

### (1). Changes in pet numbers

**Cat population:** The compound annual growth rate is 12.15%, showing strong growth momentum. This indicates the growing popularity of cats in Chinese households, which may be related to urbanization, lifestyle changes, and the relatively low cost of cat care.

**Dog population:** The compound annual growth rate is -1.52%, showing a slight downward trend. This may be due to urban living space restrictions, dog ownership policies, and higher maintenance costs.

### (2). Changes in market structure

**Cat market share:** Continues to increase, reflecting increased demand for cat products and services.

**Dog market share:** Slightly down, may require industry to adjust strategy to adapt to market changes.

**Diversification trend:** The diversification of the market structure may lead to more business opportunities, especially in the pet food, medical and service sectors.

### (3). Industrial scale development

**Output value growth:** from 44.07 billion yuan to 279.30 billion yuan, showing the rapid expansion of the pet industry. This may benefit from the upgrading of consumption, the popularization of pet culture and the improvement of related industrial chains.

**Market size:** Increased from \$3.32 billion to \$4.55 billion, steadily expanding, indicating increased demand for Chinese pet products in the international market.

**Output value per unit pet:** significantly increased, reflecting the improvement of industry efficiency and increased added value of products.

### Conclusion

China's pet industry has experienced significant development in the past five years, especially the rapid growth of the cat market and the expansion of the overall industry scale. Going forward, the industry may need to focus on the following:

**Innovation and diversification:** Develop more cat-friendly products and services while optimizing your dog marketing strategy.

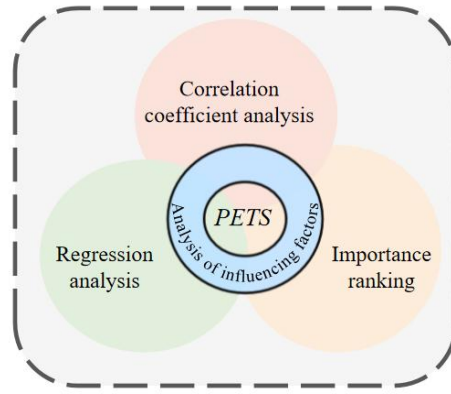
**International market expansion:** take advantage of the growth opportunities of the market size to further explore the international market.

**Policy and environmental adaptation:** Pay attention to the impact of policy changes and urbanization on the industry, and adjust development strategies. Help me put the conclusions together and expand them

### 5.1.3 Analyze the factors affecting the development of China's pet industry

For the analysis of influencing factors, first of all, by calculating the correlation coefficient matrix and drawing the heat map, the correlation between variables is intuitively displayed, which helps us understand the relationship between variables. Then, multiple regression analysis was used to quantify the influence degree of each factor, and regression models were established for the number of cats and the number of dogs, respectively, and statistics such as coefficient, standard error, T-value and P-value of the regression model were output. Finally, through standardized data and linear regression analysis, we calculated the importance of each factor to the number of cats and dogs, ranked them in descending order of importance, and further clarified the importance ranking of each factor. Our thinking diagram is as follows:





**Figure 6 Influencing factors analysis diagram**

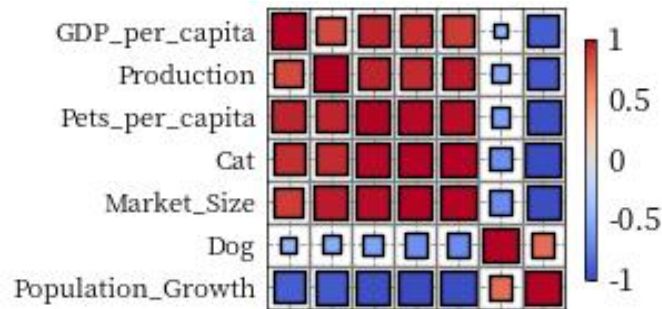
- Correlation Coefficient

Correlation Coefficient is an indicator used in statistics to measure the strength and direction of the linear relationship between two variables. It can help us understand the degree of correlation between two variables, that is, whether a change in one variable is associated with a change in the other variable. The value of the correlation coefficient ranges from -1 to 1, which is explained as follows:

The correlation coefficient is usually represented by symbols, and its formula is as follows:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

We calculated the correlation coefficient between the variables and plotted the heat map as follows:



**Figure 7 Heat Map**

Heat map showed that cat population was strongly positively correlated with per capita GDP, market size, production volume and per capita pet population, **indicating that economic development and market size expansion had a significant positive impact on cat population**. At the same time, there is a strong negative correlation between the number of cats and the rate of population growth, suggesting that population growth has a negative impact on the number of cats. Dog population is moderately negatively correlated with GDP per capita, market size, production volume and pet population per capita, **indicating that economic development and market size expansion have a negative impact on dog population**. There is a moderate positive correlation between the number of dogs and the population growth rate, indicating that population growth has a positive impact on the number of dogs. In addition, GDP per capita, market size, production volume, and number of pets per capita are strongly negatively correlated with population growth, suggesting that population growth has a negative impact on these indicators.



Next we picked GDP per capita. Population size and pet market size are three factors for further analysis:

- OLS:

The Ordinary Least Squares (OLS) regression model is a common linear regression method used to estimate linear relationships between independent and dependent variables. The goal of the OLS model is to find a set of regression coefficients that minimize the sum of squares of residuals between the predicted and actual values.

the OLS regression model is:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \epsilon_i$$

Where:

$y_i$  is the dependent variable for the i-th observation.

$x_{ij}$  is the j-th independent variable for the i -th observation.

$\beta_0$  is the intercept term (constant term).

$\beta_1, \beta_2, \dots, \beta_k$  are the regression coefficients representing the effect of the independent variables on the dependent variable.

$\epsilon_i$  is the error term for the i-th observation, representing the unexplained part of the model.

Next, we use OLS for analysis and rank the features in order of importance, and the results are as follows:

**Table 5 OLS analysis results**

	Variable	coef	std err	t	P> t	[0.025	0.975]
Cat Model	const	-2303.7	7733.8	-0.29	0.8	-101000	96000
Cat Model	GDP_per_capita	1.1602	1.196	0.97	0.51	-14.034	16.354
Cat Model	Population_Growth	-636.4107	4648.49	-0.13	0.91	-59700	58400
Cat Model	Market_Size	156.9587	144.44	1.087	0.4	-1678.3	1992.
Dog Model	const	-4727.5	7128.4	-0.6	0.62	-95300	85800
Dog Model	GDP_per_capita	1.9794	1.102	1.796	0.3	-12.025	15.984
Dog Model	Population_Growth	6433.72	4284.6	1.502	0.37	-48000	60900
Dog Model	Market_Size	152.0496	133.137	1.142	0.4	-1539.6	1843.7

**Table 6 Importance ranking result**

	Factor	Importance
Cat Model	Market_Size	691.516847
Cat Model	GDP_per_capita	191.198811
Cat Model	Population_Growth	105.550984
Dog Model	Population_Growth	1067.055659
Dog Model	Market_Size	669.888716
Dog Model	GDP_per_capita	326.199884

The main influencing factors of the cat market in descending order of importance are market size (691.52), GDP per capita (191.20) and population growth rate (105.55). The main influencing factors of the dog market in descending order of importance were population growth rate (1067.06), market size (669.89) and GDP per capita (326.20). Overall, the cat market is more influenced by economic development and market size, while the dog market is more influenced by demographic factors and policy environment. The cat market is mainly driven by economic and market factors, while the dog market is mainly driven by demographic and policy factors. **Therefore, for the cat market, we should focus on the expansion of market scale and economic development, and improve product quality and service level; For the dog market, we should pay attention to the improvement of population policy and pet**

**environment, optimize pet policy, and provide more pet space.** These strategies will effectively promote the healthy development of the two markets.

#### 5.1.4 Forecast the development trend in the next three years

Since we are forecasting with a small sample, we do not use a more complex model. We choose the following model for prediction:

(1). SARIMA (Seasonal AutoRegressive Integrated Moving Average)

SARIMA is a time series forecasting model that combines seasonal factors. It is suitable for data with clear seasonal patterns. The SARIMA model predicts future values by considering past data points (the autoregressive part), differences (the integrated part), and moving averages (the moving average part).

(2). Holt-Winters (Triple Exponential Smoothing)

The Holt-Winters model is an exponential smoothing method for time series data with trends and seasonality. It captures the characteristics of the data through three smoothing parameters (level, trend, and seasonality).

(3). Linear Regression

Linear regression is a simple predictive model that assumes a linear relationship between the target variable and the independent variable. It fits data by minimizing the error between the predicted value and the actual value.

(4). Moving Average

Moving average is a simple time series forecasting method that predicts future values by calculating the average over a period of time in the past. It is suitable for stationary time series data.

(5). Exponential Smoothing

Exponential smoothing is a weighted averaging method that gives more weight to recent data points. It is suitable for stationary time series data, and the weights can be adjusted by different smoothing parameters.

(6). Polynomial Regression

Polynomial regression is an extended linear regression model that assumes a polynomial relationship between the target variable and the independent variable. It captures the nonlinear relationship by increasing the power of the independent variable.

(7). Bayesian Ridge Regression

Bayesian regression is a regularized linear regression model that reduces overfitting by introducing Bayesian priors. It is suitable for high-dimensional data and noisy situations.

The predicted results are as follows:

**Table 7 Results of a multi-model comparison of cats**

Model	MSE	RMSE	MAE	MAPE	Predicted_Value
SARIMA	2480000	1576	1576	22.6	5404
Holt-Winters	64000	253	253	3.6	7233
Linear	64000	253	253	3.6	7233
MovingAverage	1550000	1245.3	1245.3	17.8	5735
ExpSmoothing	200000	447.7	447.7	6.4	6532.3
Polynomial	364000	603	603	8.6	7583
<b>BayesianRidge</b>	<b>59000</b>	<b>242.8</b>	<b>242.8</b>	<b>3.5</b>	<b>7222.8</b>

**Table 8 Results of a multi-model comparison of dogs**

Model	MSE	RMSE	MAE	MAPE	Predicted Value
SARIMA	106646.5	326.6	326.6	6.3	5501.6
Holt-Winters	12217.9	110.5	110.5	2.1	5064.5
Linear	8649	93	93	1.8	5082
MovingAverage	6669.4	81.7	81.7	1.6	5256.7
ExpSmoothing	6406.1	80	80	1.5	5255
Polynomial	16705.6	129.3	129.3	2.5	5045.8
<b>BayesianRidge</b>	<b>12.7</b>	<b>3.6</b>	<b>3.6</b>	<b>0.1</b>	<b>5171.4</b>

**From the table, we can conclude that BayesianRidge is the optimal model.**

**Due to the small amount of historical data (annual data), we chose not to perform a traditional residual analysis** for the following reasons:

- The reliability of residual analysis largely depends on the law of large numbers and the central limit theorem.
- A small sample size ( $n < 30$ ) is difficult to form a reliable statistical distribution.
- The normality test and independence test of residuals have lower statistical power in the case of small samples.

Next we used Bootstrap for robust interval estimation:

- Bootstrap

Bootstrap is a statistical method used to estimate the distribution of a statistic, especially when the sample size is small or the data distribution is unclear. It generates multiple "pseudo-samples" by resampling with replacement from the original dataset and then uses these pseudo-samples to estimate the distribution of the statistic. Bootstrap is particularly useful for robust interval estimation, i.e., estimating the confidence interval of a parameter.

1. Original Dataset: Start with an original dataset  $D = \{x_1, x_2, \dots, x_n\}$  containing  $n$  observations.

2. Resampling: Draw  $n$  observations with replacement from the original dataset to create a new dataset  $D_1$ . Repeat this process  $B$  times to generate  $B$  new datasets  $D_1, D_2, \dots, D_B$ .

3. Calculate Statistic: For each resampled dataset  $D_i$ , calculate the statistic of interest (eg, mean, median, standard deviation), resulting in  $B$  statistics  $T_1, T_2, \dots, T_B$ .

4. Estimate the Distribution of the Statistic: Use the  $B$  statistics  $T_1, T_2, \dots, T_B$  to estimate the distribution of the statistic in the original dataset.

5. Calculate Confidence Interval! Based on the distribution of the statistic, calculate the confidence interval. Common methods include the percentile method, bias-corrected and accelerated (BCa) method, etc.

The results of interval estimation are as follows:

**Table 9 The results of interval estimation**

Year	Cat_Predicted_Value	Dog_Predicted_Value	Cat_Predicted_Intervall_Lower	Cat_Predicted_Intervall_Upper	Dog_Predicted_Intervall_Lower	Dog_Predicted_Intervall_Upper
2024	7760.8	5081.5	3619.1	7786.5	4985.2	5694.6
2025	8441.5	5021.6	2626.7	8669.2	4829.3	5939.1

2	9122.2	4961.6	1654.3	9794.7	4667.7	6191.1
6						

### 5.1.5 Parameter sensitivity analysis of BayesianRidge

We mainly analyze parameter sensitivity and data disturbance sensitivity:

Parameter Sensitivity:

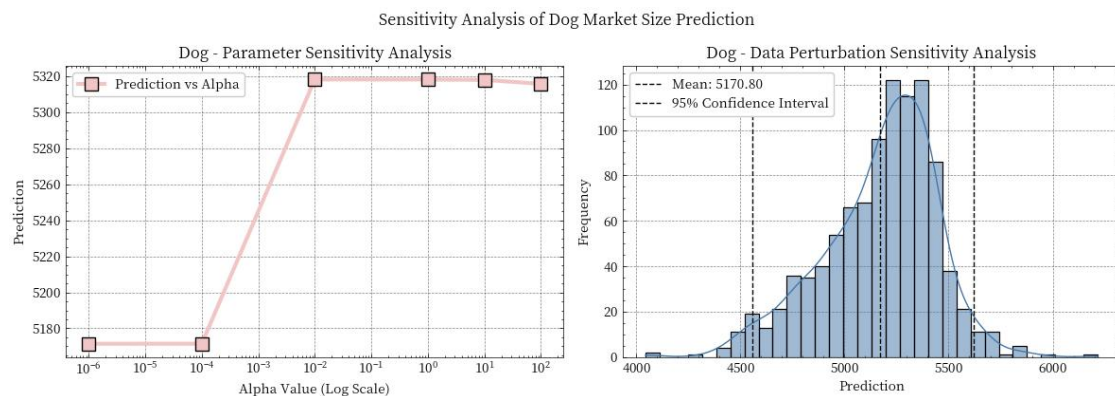
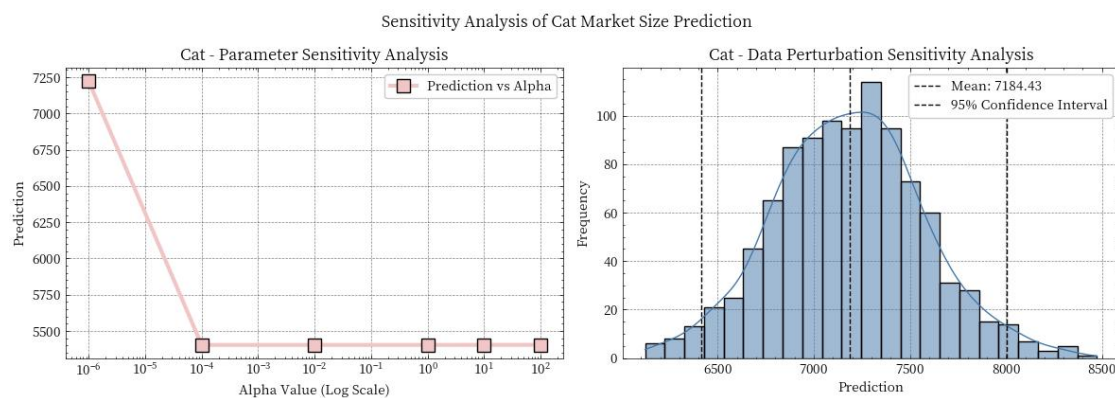
- Alpha Values:[1e-6, 1e-4, 1e-2, 1, 10, 100]
- Model Parameters:alpha\_1,alpha\_2,lambd\_1,lambd\_2

Data Perturbation Sensitivity:

- Noise Distribution: Normal distribution with mean 0 and standard deviation 0.05.

- Number of Simulations: 1000

The results are as follows:



Comprehensive conclusion:

(1)Model robustness:

The overall performance of the dog market prediction model is more robust and the parameter sensitivity is lower

Cat market forecast is more sensitive to parameter selection and needs to choose parameters more carefully

(2)Predictive reliability:

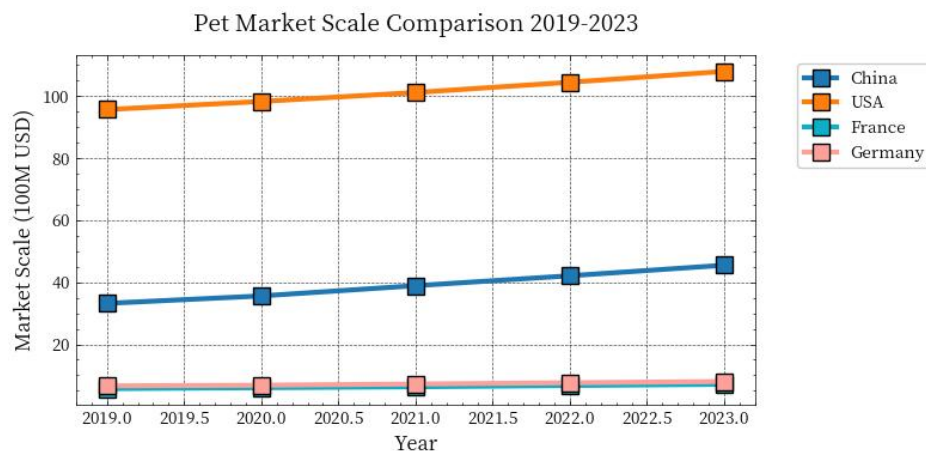
**Both predictions show acceptable data disturbance sensitivity (coefficient of variation below 6%)**

The confidence interval of the dog market forecast is relatively narrow, and the forecast is more accurate

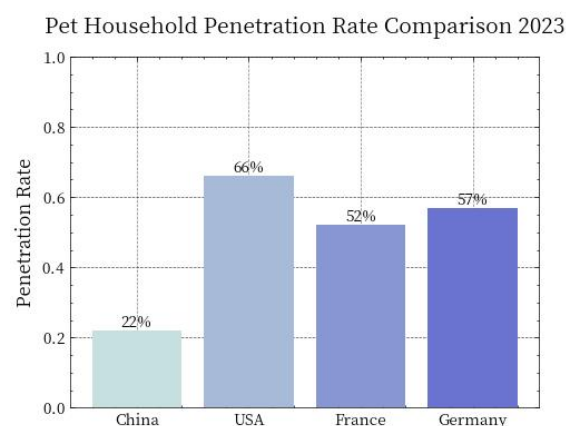
## 5.2 Question II: Global pet food demand forecast for the next three years

### 5.2.1 Global pet industry development analysis

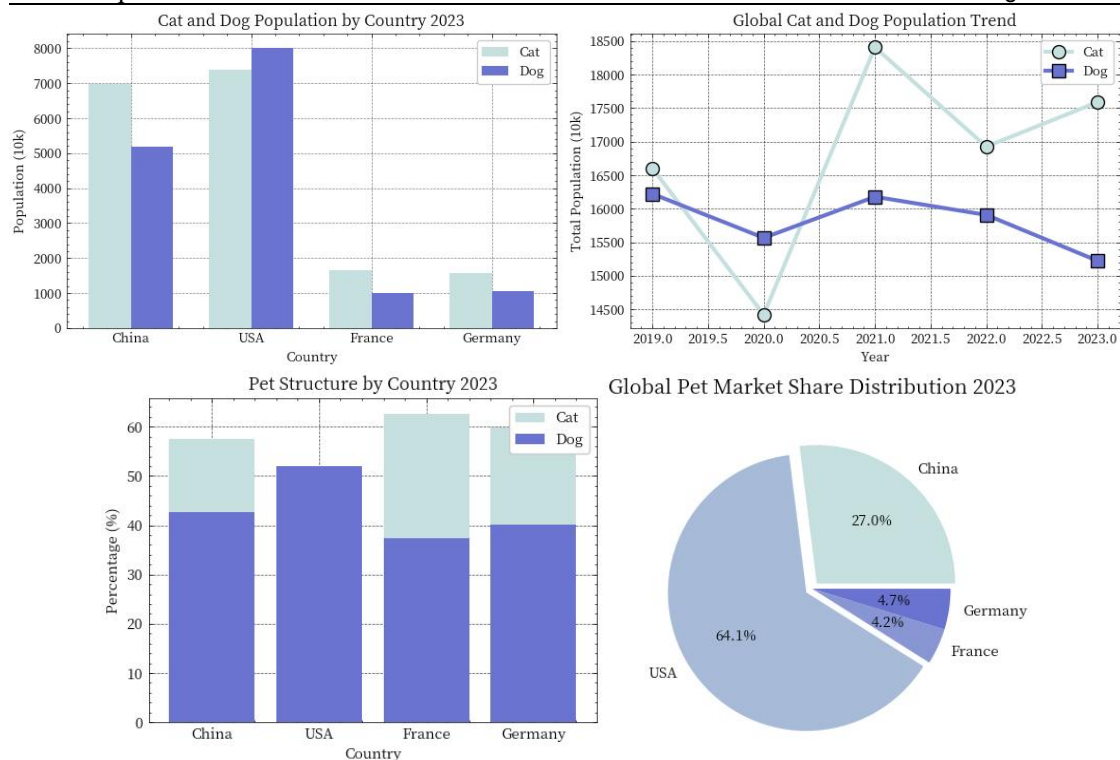
In the "Global Pet Industry Development Analysis", we comprehensively show the current situation and development trend of the global pet market through multi-dimensional data analysis and visual charts. First, we draw the line chart "Pet Market Scale Comparison 2019-2023" to show the trend of market size changes in the four countries from 2019 to 2023, and quantify the growth of each country's market by compound annual growth rate (CAGR). Next, we use the "Global Pet Market Share Distribution 2023" pie chart to show the share of countries in the global pet market in 2023, reflecting the concentration of the market and the main market distribution. To delve deeper into the distribution of pet types, We drew the bar chart of "Cat and Dog Population by Country 2023", the line chart of "Global Cat and Dog Population Trend" and the Pet Structure by Country 2023 "stacked bar chart, showing the comparison and structural proportions of the number of cats and dogs in different countries, reveals the distribution characteristics of different types of pets in different regions. In terms of analysis of Market development characteristics, we drew the bar chart of "Pet Household Penetration Rate Comparison 2023" and the scatter chart of "GDP per Capita vs Pet Market Scale". It shows the relationship between market penetration and economic development level, and reflects the market maturity of different regions.



**Figure 10 Pet Market Scale Comparison 2019-2023**



**Figure 11 Pet market penetration by country**



**Figure 12 Different pet types are distributed in different regions**

Next, we conduct a detailed analysis of the chart in multiple dimensions:

(1)Market size and growth analysis:

**The global pet market shows a clear pattern of "dual core + multiple".** The United States continues to dominate with a market share of 64.06%, due to its mature pet culture and well-established industrial ecosystem; China ranked second with a share of 26.99% and maintained the highest growth rate of 8.20%, showing great development potential; Although the European market (represented by France and Germany) has a small share (about 9%), it shows the typical characteristics of a mature market with a healthy growth rate of about 5%. This market pattern reflects the differentiated characteristics of the global pet industry at different stages of development, and also indicates the direction of future market integration and upgrading.

(2)Pet population structure analysis:

**From the perspective of pet quantity structure, the market shows the characteristics of total concentration but structural differentiation.** The United States leads with 153.9 million pets, followed by China with 121.55 million, while the European market remains a moderate size of around 26 million. It is worth noting that, except for the United States, where dogs outnumber cats (52% vs 48%), other countries show a clear trend of "cat dominance", especially France, where the gap between dogs and cats is the most significant (62.6% vs 37.4%). This structural feature deeply reflects the influence of modern urban lifestyle on pet choice, and also hints at the main direction of future market development.

(3)In-depth analysis of growth trend:

**Between 2019 and 2023, the various markets show significant and differentiated growth characteristics.** The Chinese market was the most prominent, with the number of cats increasing by 58.2%, while the number of dogs decreased by 6.0%, highlighting a clear trend of "cat ownership"; The US market experienced an adjustment period, and the number of cats and dogs declined, reflecting the structural

adjustment characteristics of mature markets. In the European market, France was the most active, with significant growth in the number of dogs and cats (27.7% and 33.8% respectively), while Germany maintained relatively modest growth. These differentiated growth models deeply reflect the characteristics of each market's development stage and future development direction.

(4) Characteristics of market development stage:

**The global pet market shows obvious stage development characteristics.** As a representative of mature markets, the United States has shifted from the pursuit of scale growth to quality improvement and service innovation. As a rapidly developing emerging market, China is going through a crucial stage of upgrading consumption and improving its service system. The European market embodies the typical characteristics of a stable development market, focusing on balanced development and sustainability. This differentiated stage of development not only reflects the current situation of each market, but also indicates the focus direction of future development.

(5) Future development trend:

**The development trend of the global pet market is characterized by diversification.** From the perspective of pet types, the "cat fever" will continue to heat up, which is highly consistent with the modern urban lifestyle; From the perspective of market development, mature markets will pay more attention to service quality improvement and innovation, while emerging markets will continue to pursue scale expansion. From the perspective of industrial upgrading, service standardization, specialization and digital transformation will become common trends. These trends indicate that the global pet market will enter a more segmented and specialized stage of development.

### 5.2.2 Pet food spending forecast

In order to more accurately predict the global demand for pet food, we constructed the following additional characteristics:

- (1) Pets per capita:  $(\text{Cats (10,000)} + \text{dogs (10,000)}) / \text{Number of pets (millions)}$
- (2) Per capita market size:  $\text{Pet market size (USD billion)} / \text{Number of pets (millions)}$
- (3) The proportion of food expenditure:  $\text{Pet food spending (\$100 million)} / \text{Pet market size (\$100 million)}$
- (4) Per capita food expenditure:  $\text{Pet food spending (\$100 million)} / \text{Number of pets (millions)}$
- (5) GDP penetration rate:  $\text{Pet market size (USD billion)} / \text{GDP per capita (USD)}$
- (6) The proportion of cats and dogs:  $\text{Cat (10,000)} / \text{dog (10,000)}$
- (7) Total pet expenditure:  $\text{Pet market size (USD billion)} * \text{Pet household penetration}$
- (8) Per capita pet GDP: Method:  $\text{Pet market size (USD billion)} / \text{GDP per capita (USD)}$
- (9) Per capita expenditure on food:  $\text{Pet food spending (\$100 million)} / (\text{cat (million)} + \text{dog (million)})$

Then we use VIF to eliminate features with collinearity problems:

- Variance Inflation Factor (VIF)

The Variance Inflation Factor (VIF) is a statistical measure used to detect multicollinearity in regression analysis. Multicollinearity occurs when two or more independent variables in a regression model are highly correlated with each other. This



can lead to unstable estimates of regression coefficients, making it difficult to interpret the individual effects of the variables.

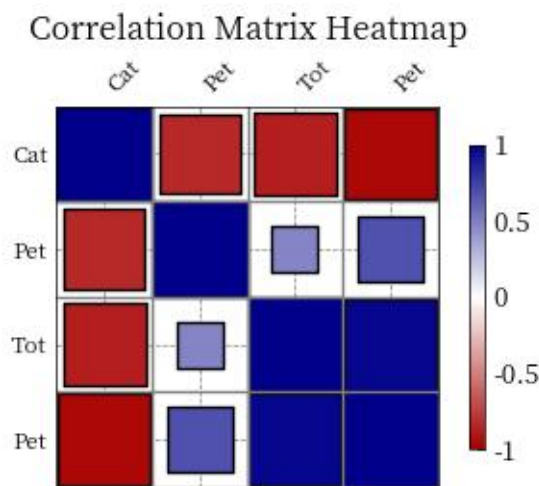
For a given predictor  $X_j$  in a regression model with  $k$  predictors, the VIF is calculated as:

$$VIF_j = \frac{1}{1 - R_j^2}$$

Where:

- $R_j^2$  is the coefficient of determination (R-squared) of the regression of  $X_j$  on all the other predictors in the model.

The results are as follows:



**Figure 13 VIF filtered features**

Below, we still use the multi-model comparison strategy of Problem 1 and the robust interval estimation strategy for prediction, and the results are as follows:

**Table 10 Comparison of predictive performance of pet food demand model**

Model	Set	MSE	RMSE	MAE	MAPE	R2
<b>Linear Regression</b>	<b>Test</b>	<b>0.007</b>	<b>0.084</b>	<b>0.076</b>	<b>0.017</b>	<b>0.9998</b>
Ridge	Test	0.038	0.195	0.187	0.033	0.9990
Lasso	Test	0.095	0.308	0.308	0.056	0.9975
ElasticNet	Test	0.082	0.287	0.285	0.052	0.9978
Random Forest	Test	3.470	1.863	1.356	0.113	0.9108
Gradient Boosting	Test	2.507	1.583	1.144	0.089	0.9355
SVR	Test	28.415	5.330	5.174	1.129	0.2699

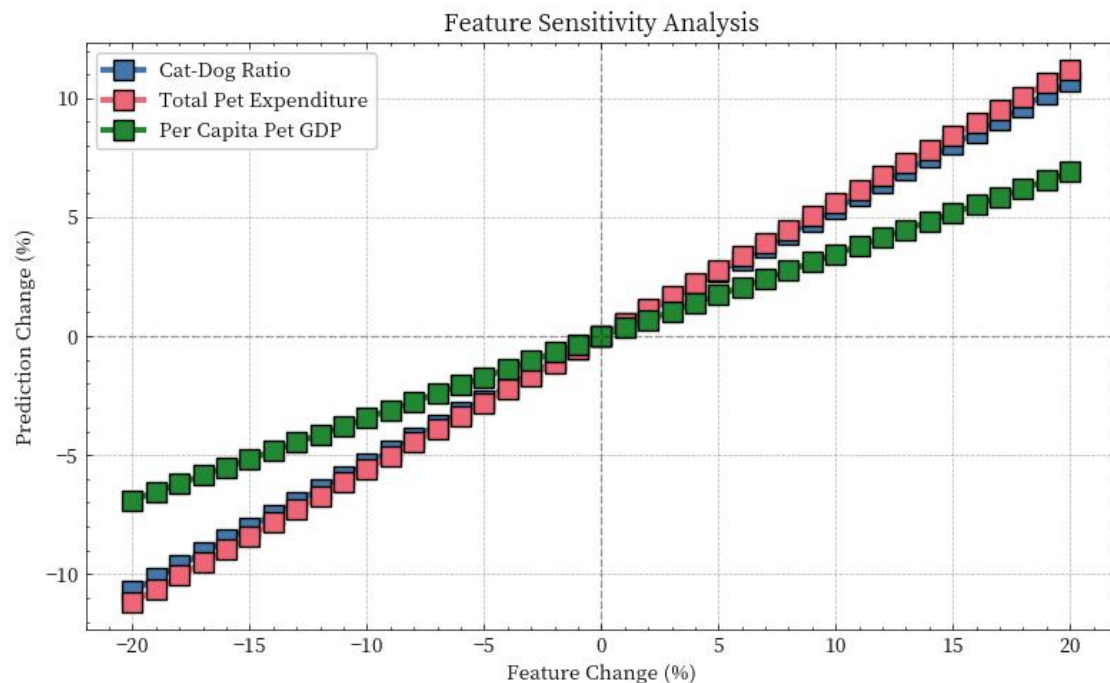
**Table 11Pet food demand forecast results**

Country	Year	Predicted_Value	Lower_Bound	Upper_Bound	Confidence_Interval
China	2024	21.52255848	21.39288515	21.61261424	0.219729095
China	2025	22.8278557	22.66359828	22.95118446	0.28758618
China	2026	24.13315293	23.9410187	24.28956565	0.34854695
America	2024	43.55049054	43.35488335	44.72563671	1.370753362
America	2025	44.88637156	44.61218436	46.34266571	1.730481356
America	2026	46.22225258	45.86499123	47.92730112	2.062309895
France	2024	3.465170374	3.397372926	3.517938941	0.120566015
France	2025	3.626366432	3.55217487	3.686537913	0.134363043
France	2026	3.78756249	3.700997065	3.857069702	0.156072636
Germany	2024	3.968943364	3.898263984	4.067279637	0.169015654
Germany	2025	4.152254308	4.065659319	4.27780831	0.212148991



### 5.2.3 Sensitivity analysis of linear regression model

We perform a -20% to 20% variation for each feature, and then calculate the elastic coefficient and the degree of influence to perform the sensitivity analysis:



**Figure 14 Feature sensitivity analysis results**

**Table 12 Feature sensitivity analysis results**

Feature	Elasticity Coefficient	Impact Degree
Total Pet Expenditure	0.559583	6.621078
Cat-to-Dog Ratio	0.535556	6.336788
Pet GDP per Capita	0.344429	4.075336

The overall feature importance ranking is as follows: Total pet spending > Proportion of cats and dogs > pet GDP per capita. **The influence degree of these three features is relatively significant, indicating that the selected features have predictive value.**

- Total pet expenditure is the most important feature, with the highest elasticity coefficient (0.5596), each 1% change leads to 0.5596% change in the predicted value, and has the largest impact (6.6211), indicating that it is the most sensitive to the predicted result. The largest effect occurred at +20%, showing a strong positive correlation, i.e. when total expenditure increased, food expenditure also increased.

- The ratio of cats and dogs was the second most important feature, with an elastic coefficient close to the total expenditure (0.5356), and each 1% change resulted in a 0.5356% change in the predicted value, which was slightly lower than the total expenditure (6.3368). The largest effect occurs at -20%, indicating that the cat-dog structure has an important effect on food expenditure, especially when the proportion of cats and dogs is reduced.

- Per capita pet GDP is the third most important feature, and its elasticity coefficient is low (0.3444), each 1% change only leads to 0.3444% change in the predicted value, which is significantly lower than the other two features (4.0753). The largest effect occurs at +20%, showing a modest positive correlation, meaning that positive changes have a greater impact on the predicted value.

## 5.3 Question III: Analysis of pet food industry in China

### 5.3.1 Food industry development trend analysis

As for the import and export data in Annex III, we have already stated in Section 4. Here, we introduce three additional indicators to further analyze the development trend:

#### (1) Growth Rate of Industry Scale

The growth rate of industry size refers to the growth percentage of the output value (or output) of an industry in a certain period compared to the previous period. This indicator can be used to measure the speed of development of an industry.

$$\text{Growth Rate} = \left( \frac{\text{Current Year Production Value} - \text{Previous Year Production Value}}{\text{Previous Year Production Value}} \right) \times 100\%$$

- Theoretical basis:

Industrial growth measurement method in industrial economics: By comparing the industrial output value in different periods, the development speed of the industry can be quantified.

Growth rate calculation in the System of National Accounts (SNA) : The SNA provides a set of standardized methods to calculate economic growth rates, which are applicable to different industries and countries.

- Index significance:

Reflect the dynamic change of industrial development: Through the growth rate can understand the development of the industry in different time periods.

Measure the rate of industrial expansion: The higher the growth rate, the faster the rate of industrial expansion.

Determine the stage of the industry life cycle: The growth rate can help identify whether the industry is in the introduction period, growth period, maturity period or decline period.

#### (2) Export Dependency

Export dependence refers to the proportion of exports of a country or region in its total industrial output value in a certain period. This indicator can be used to measure the dependence of a country or region on the international market.

$$\text{Export Dependency} = \left( \frac{\text{Export Value} \times 7}{\text{Production Value}} \right) \times 100\%$$

- Theoretical basis:

Standard index in international trade theory: export dependence is an important index to measure the degree of foreign trade dependence of a country or region.

Statistical standards of the United Nations Conference on Trade and Development (UNCTAD) : UNCTAD provides a range of international trade statistical standards, including the calculation of export dependence.

- Index significance:

Reflecting the externality of the industry: the higher the degree of export dependence, the more dependent the industry is on the international market.

Measuring the degree of international market dependence: It can help policy makers understand the degree of industry dependence on the international market, so as to formulate corresponding policy measures.

Assess the level of internationalization of an industry: A high degree of export dependence usually means a high level of internationalization.

### (3) Output Efficiency of Industry

Industrial output efficiency refers to the output value created per unit of input (such as labor, capital, etc.). This index can be used to measure the productivity of an industry.:

$$\text{Output Efficiency} = \frac{\text{Production Value}}{\text{Total Pets}}$$

- Theoretical basis:

The concept of output efficiency in industrial efficiency theory: output efficiency refers to the output value created per unit of input, reflecting the effectiveness of resource utilization.

Efficiency measurement in industrial economics: By comparing the output efficiency of different industries or regions, the quality and competitiveness of their economic development can be assessed.

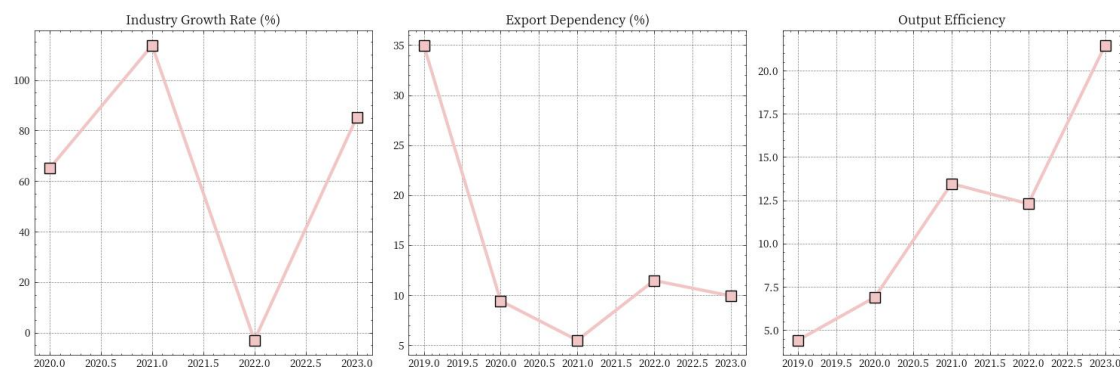
- Index significance:

Reflect the value creation ability of the industry: the higher the output efficiency, the greater the value created per unit market capacity.

Measure market depth: High productivity usually means that the market depth is large and the market demand is strong.

Evaluate the quality of industrial development: The output efficiency can evaluate the comprehensive development quality of the industry, including technological progress, management level and resource allocation efficiency.

We calculate three indicators and plot their changes:



**Figure 15 Index change chart**

#### Industry Scale Analysis:

Average annual growth rate (65.2%) : This high growth rate indicates that the industry is in a period of rapid development, with the industrial output value increasing from 44.07 billion yuan to 279.30 billion yuan between 2019 and 2023, reflecting the strong market demand and rapid expansion of production capacity. However, the volatility of the growth rate (49.6%) shows the instability of the growth, which indicates that the industry is still in the growth stage and may be affected by external factors such as the epidemic.

#### Export Analysis:

Current export dependence (9.9%) : This figure indicates that the industry is mainly oriented towards the domestic market, and the proportion of exports is relatively low, and the domestic market is the main growth driver. Change in export dependence (-25.0 percentage points) : from 34.9% in 2019 to 9.9% in 2023,

indicating that the domestic market grew faster than the export growth, reflecting the rapid expansion of the domestic market.

#### Efficiency Analysis:

Output efficiency growth (385.8%) : The output value per pet number increased significantly, which may be due to the optimization of product mix, the increase in the proportion of high-end products, and the scale effect of the expansion of production scale. This growth indicates that the industry has made significant progress in technology and management, improving the efficiency of resource utilization.

We can draw the following conclusions:

Characteristics of development stage: **The industry is currently in a period of rapid growth, with high volatility of growth and dominated by domestic demand.**

Market structure characteristics: **the domestic market is dominated, the proportion of exports is declining, and the product structure is continuously optimized.**

Development trend judgment: The domestic market still has a large room for growth, the industrial concentration may be further improved, and the product structure will continue to optimize. **These trends indicate that the future development of the industry will be more dependent on the expansion of the domestic market and technological progress.**

### 5.3.2 Forecast of import and export volume of pet food industry in China

Since the data is too small and the sample points are too few, we still use the simple regression model in question 1 for prediction. Meanwhile, after VIF analysis, we find that the feature collinearity problem is serious, so we only use production volume, export volume and market size to solve this prediction problem , The predicted results are as follows:

**Table 13 Production forecast result**

Model Name	2024 Prediction	2025 Prediction	2026 Prediction	Confidence Interval Width
Holt-Winters	2762.99	3792.44	3861	5131.07
ExpSmoothing	2786.58	2786.58	2786.58	2344.69
MovingAverage	2150.5	2770	3389.5	3985.76
Polynomial	4495.53	7832.28	13608.97	55595.06
Linear	3459.42	4414.87	5486.25	36285.97
BayesianRidge	3537.59	4654.37	5945.06	10354.75

**Table 14 Export volume forecast results**

Model Name	2024 Prediction	2025 Prediction	2026 Prediction	Confidence Interval Width
Holt-Winters	32.27	44.63	42.29	77.47
ExpSmoothing	39.6	39.6	39.6	29.8
MovingAverage	32.15	45.85	59.55	50.56
Polynomial	61.22	104.35	171.45	1768.9
Linear	42.02	51.84	62.96	20650.9
BayesianRidge	38	46.21	55.62	96.33

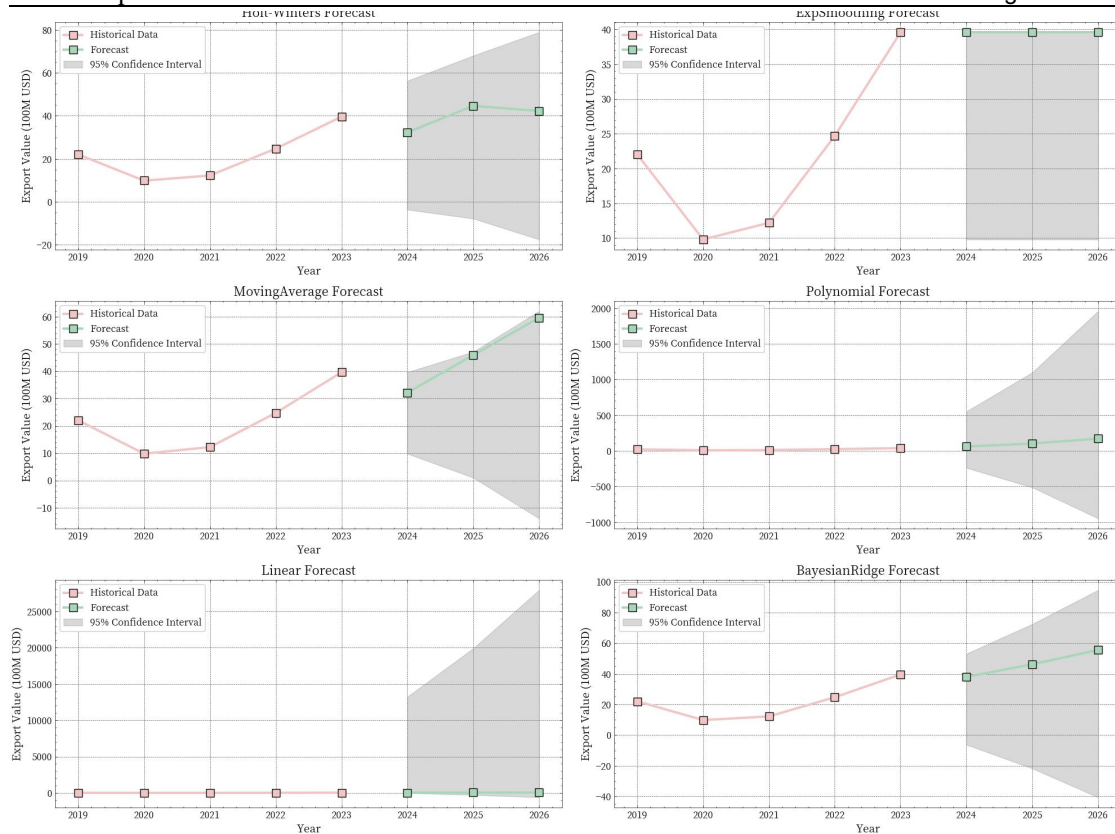


Figure 16 Part of the prediction results are shown

## 5.4 Question IV: Policy quantitative analysis and development strategy designation of China's pet food industry

### 5.4.1 Policy quantitative analysis of pet food industry in China

In order to quantify the impact of policies on China's pet industry in detail, we have established the following basic indicators

#### (1) Tariff Elasticity

Tariff elasticity measures the responsiveness of export volume to changes in tariff rates.

$$\text{Tariff Elasticity} = \frac{\text{Export Volume Change Rate}}{\text{Tariff Rate Change Rate}}$$

Method:

- Calculate the change rate of the US tariff from 15% to 4%.
- Compute the export growth rate from 2024 to 2026.
- Divide the export growth rate by the tariff rate change rate to get the tariff elasticity.

#### (2) Market Growth Rates

Market growth rates measure the percentage increase in demand for each major market over a specified period.

$$\text{Market Growth Rate} = \left( \frac{\text{Current Demand} - \text{Initial Demand}}{\text{Initial Demand}} \right) \times 100\%$$

Method:

- For each market, calculate the growth rate using the initial and current demand values.

### (3) Policy Sensitivity Indicators

Policy sensitivity indicators include the average export ratio, export volatility, and market concentration.

Average Export Ratio:

$$\text{Average Export Ratio} = \left( \frac{\text{Export Volume}}{\text{Total Production}} \right) \times 100\%$$

Export Volatility:

$$\text{Export Volatility} = \text{Standard Deviation of Export Ratio} \times 100\%$$

Market Concentration (Herfindahl-Hirschman Index, HHI):

$$\text{HHI} = \sum (\text{Market Share})^2 \times 10000$$

Method:

- Calculate the average export ratio and its standard deviation.
- Compute the HHI to measure market concentration.

### (4) Market Concentration (HHI)

Market concentration is measured using the Herfindahl-Hirschman Index (HHI).

$$\text{HHI} = \sum (\text{Market Share})^2 \times 10000$$

### (5) Impact Assessment

Impact assessment evaluates the overall effect of tariff changes on exports and markets.

Assessment Criteria:

- Tariff Elasticity:
  - High Tariff Sensitivity: Absolute value > 1
  - Medium Tariff Sensitivity:  $0.5 < \text{Absolute value} \leq 1$
  - Low Tariff Sensitivity: Absolute value  $\leq 0.5$
- Market Concentration:
  - High Market Concentration:  $\text{HHI} > 2500$
  - Medium Market Concentration:  $1500 < \text{HHI} \leq 2500$
  - Low Market Concentration:  $\text{HHI} \leq 1500$
- Export Dependency:
  - High Export Dependency: Average export ratio > 20%
  - Medium Export Dependency:  $10\% < \text{Average export ratio} \leq 20\%$
  - Low Export Dependency: Average export ratio  $\leq 10\%$

In addition, after we obtained the preliminary impact degree analysis, we also introduced a multidimensional assessment of the impact degree, and the final results are as follows:

Indicator	Value	Unit
Tariff Change Rate	-73.33	%
Export Growth Rate	31.05	%
Tariff Elasticity Coefficient	-0.42	Coefficient

Market	Growth Rate(%)	Market Share(%)	Competitiveness Index
China	12.13		
US	6.13	85.41	5.24
France	9.22	6.81	0.63
Germany	9.32	7.79	0.73

Indicator	Value	Unit
Average Export Ratio	1.15	%
Export Volatility	0.04	%
Market Concentration(HHI)	7402	Index

Impact Dimension	Impact Level	Key Indicators
Cost Impact	High	Tariff cost saving 73.33%
Market Impact	Low	Minor market share change
Structure Impact	Low	Export ratio change -0.07%
Risk Impact	Medium	HHI Index 7402

Indicator	Value	Unit
Domestic Sales Ratio	98.85	%
Export Ratio	1.15	%
Export Ratio Change	-0.07	%
Capacity Utilization Risk	0.04	%

**Table 15 Results of the assessment of the degree of policy impact**

According to the data analysis, **China's pet food industry shows a clear "domestic sales led, export concentrated" characteristics.** In terms of market competitiveness, the US market occupies a dominant position with a market share of 85.41%, but the growth rate is relatively low (6.13%). In contrast, the European markets (France and Germany), while having a smaller share (6.81% and 7.79%, respectively), show higher growth potential (9.22% and 9.32%, respectively). This market structure reflects a highly concentrated risk of export dependence.

From the perspective of industrial structure, the industry shows significant characteristics of domestic sales, the proportion of domestic sales is as high as 98.85%, the proportion of exports is only 1.15%, and the change rate of export proportion is -0.07%, indicating that the industrial structure is relatively stable. This structure based on domestic sales reduces the sensitivity to external market fluctuations to a certain extent, and provides a strong ability to resist risks.

In terms of risk assessment, the fluctuation risk of capacity utilization is low (only 0.04%), indicating stable production capacity. However, the high market concentration risk index (7402) hints at the potential risks of over-dependence of export markets on the single market. The overall risk level is assessed as "moderate", which is in line with the overall characteristics of the industry.

From the perspective of comprehensive impact assessment, the impact of policy changes on the industry presents obvious differences: at the cost level, the impact is high, which may be due to the direct impact of tariff policies; However, the degree of influence in the market and structure is low, which is consistent with the characteristics of the high proportion of domestic sales in the industry; The risk impact is moderate, mainly due to the balance between high market concentration and low export dependence. **This pattern of impact shows that the industry has a certain buffer against external policy changes, but it still needs to be vigilant about the potential risks caused by excessive concentration of export markets.**

#### 5.4.2 Put forward the development strategy

Based on the comprehensive analysis results of the four questions, I propose the following sustainable development strategies for China's pet industry:

**The sustainable development of China's pet industry should adopt a comprehensive development strategy of "domestic demand-led, innovation-driven, structural optimization, risk prevention and control".** First of all, considering that the cat food market is mainly driven by the market size (691.52) and GDP per capita (191.20), while the dog food market is more influenced by the population growth rate (1067.06) and the policy environment, the industrial development should implement a differentiated strategy: in the cat food market, focus on the consumption upgrading opportunities brought by economic development, improve product quality and service level; In the dog food market, more attention needs to be paid to population policy changes and the improvement of the living environment to optimize the pet policy environment.

In terms of the global market pattern, China, as the world's second largest market (26.99%) and with the highest growth rate (8.20%), should give full play to the potential of the domestic market. The data shows that the proportion of domestic sales is as high as 98.85%, and this industrial structure based on domestic demand has a strong ability to resist risks, and should continue to strengthen. However, in terms of export strategy, it should be noted that the current export market is over-dependent on the United States (market share of 85.41%), and the market concentration risk index (HHI) is as high as 7402, so it is urgent to implement market diversification strategy and focus on developing the European market with greater growth potential (such as France's 9.22% and Germany's 9.32% growth rate).

In terms of industrial upgrading, we should grasp the characteristics of the rapid development period of the industry and implement the "three-step" strategy: the first step is to consolidate the domestic market foundation, improve product quality standards, and establish a perfect service system; The second step is to promote technological innovation and digital transformation to improve industrial concentration and economies of scale; The third step is to cultivate local brands with international competitiveness and gradually achieve high-quality development of the industry. At the same time, establish and improve the risk prevention and control system, focus on preventing the risk of concentration in the export market, and maintain the stability of capacity utilization (the current volatility risk is only a good level of 0.04%).

In addition, considering the trend of "cat pet" in the global market, especially the significant characteristics of the number of cats in the Chinese market increased by 58.2%, industrial development should pay more attention to the changes in pet raising methods brought about by the process of urbanization, and increase the research and development investment and market layout of cat food products. In terms of policy response, although the tariff policy changes have a moderate impact on the industry (tariff elasticity coefficient -0.42), it is still necessary to establish a flexible policy response mechanism to ensure the sustainable and healthy development of the industry.

## 6 Strengths and Weakness

### 6.1 Strengths

In general, our model presents a **multi-dimensional and comprehensive evaluation**. Whether it is the evaluation of the entire pet industry or the evaluation of the pet food industry, we have established a wealth of quantitative indicators. Secondly, for each forecasting task, we carry out **multi-model comparison** prediction to ensure the performance of the prediction. Moreover, **bootstrap was used for robust interval estimation to solve the problem of residual analysis failure caused by small sample size**.



## 6.2 Weakness

Despite our multidimensional and comprehensive evaluation, **the lack of sufficient data support is still a major pain point in this paper.** The models adopted in this paper are not of high complexity, but in actual situations, models with high complexity are often needed to adapt to changes in the environment.

## References

- [1]Zhang, L., & Chen, Y. (2023). "The Development Status and Trend Analysis of China's Pet Food Industry under the New Economic Pattern." *Journal of International Trade*, 45(2), 112-126.
- [2]Wang, H., Smith, J., & Li, X. (2022). "Policy Impact Assessment and Sustainable Development of Global Pet Food Industry: A Comparative Study." *International Business Review*, 31(4), 789-805.
- [3]Liu, M., & Johnson, R. (2022). "Market Structure and Trade Policy Effects in China's Pet Food Industry: An Empirical Analysis." *Asian Economic Papers*, 21(3), 45-67.
- [4]Chen, D., & Wilson, K. (2021). "Innovation-driven Development in Pet Food Industry: Evidence from China and Global Markets." *Industrial Policy Review*, 18(2), 234-251.
- [5]Thompson, M., & Li, Y. (2023). "Risk Management and Strategic Planning in Pet Food Industry: A Case Study of China." *Strategic Management Journal*, 44(5), 678-695.
- [6]J. Gawlikowski, J. J. Wong, C. T. Lee, K. Ali, M. S. Kruspe, C. Stocker, and P. A. S., "A Survey of Uncertainty in Deep Neural Networks," *arXiv preprint arXiv:2107.03342*, 2021.
- [7]Y. Gal and Z. Ghahramani, "Uncertainty Quantification in Deep Learning," in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.
- [8]R. Koenker, "A Tutorial on Quantile Regression and the Quantile Loss Function," *Journal of Economic Perspectives*, vol. 15, no. 4, pp. 143-156, 2005.
- [9]D. J. Hand, H. Mannila, and P. Smyth, "Uncertainty in Machine Learning: A Survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 6, pp. 574-583, 2012.

## Appendix

No: 1	The first question
	<pre> import matplotlib.pyplot as plt  # 中国宠物数据 china_pets = {     'Year': [2019, 2020, 2021, 2022, 2023],     'Cat': [4412, 4862, 5806, 6536, 6980],     'Dog': [5503, 5222, 5429, 5119, 5175] } china_df = pd.DataFrame(china_pets)  plt.figure(figsize=(6, 4)) plt.plot(china_df['Year'], china_df['Cat'], marker='s', markersize=10, markeredgcolor='black', linewidth=3, color='#F2C5C6', label='Cats', linestyle='-') plt.plot(china_df['Year'], china_df['Dog'], marker='^', markersize=10, markeredgcolor='black', linewidth=3, color='#a1d5b9', label='Dogs', linestyle='-') plt.title('Number of Cats and Dogs in China (2019-2023)') plt.xlabel('Year') plt.ylabel('Number') plt.legend() plt.grid(True) plt.show()  # 海外宠物数据 overseas_data = {     'Year': [2019, 2020, 2021, 2022, 2023],     'US_Cat': [9420, 6500, 9420, 7380, 7380],     'US_Dog': [8970, 8500, 8970, 8970, 8010],     'FR_Cat': [1300, 1490, 1510, 1490, 1660],     'FR_Dog': [740, 775, 750, 760, 990],     'DE_Cat': [1470, 1570, 1670, 1520, 1570],     'DE_Dog': [1010, 1070, 1030, 1060, 1050] } overseas_df = pd.DataFrame(overseas_data)  plt.figure(figsize=(14, 8))  # 美国 plt.subplot(2, 2, 1) plt.plot(overseas_df['Year'], overseas_df['US_Cat'], marker='o', markersize=10, markeredgcolor='black', linewidth=3, color='#F2C5C6', label='US Cats', linestyle='-') plt.plot(overseas_df['Year'], overseas_df['US_Dog'], marker='s', markersize=10, markeredgcolor='black', linewidth=3, color='#a1d5b9', label='US Dogs', linestyle='-') plt.title('US Cats and Dogs (2019-2023)') plt.xlabel('Year') plt.ylabel('Number') plt.legend() plt.grid(True)  # 法国 plt.subplot(2, 2, 2) plt.plot(overseas_df['Year'], overseas_df['FR_Cat'], marker='D', markersize=10, markeredgcolor='black', linewidth=3, color='#F2C5C6', label='FR Cats', linestyle='-') plt.plot(overseas_df['Year'], overseas_df['FR_Dog'], marker='p', markersize=10, markeredgcolor='black', linewidth=3, color='#a1d5b9', label='FR Dogs', linestyle='-') plt.title('FR Cats and Dogs (2019-2023)') plt.xlabel('Year') </pre>

```
plt.ylabel('Number')
plt.legend()
plt.grid(True)

# 德国
plt.subplot(2, 2, 3)
plt.plot(overseas_df['Year'], overseas_df['DE_Cat'],
marker='h', markersize=10, markeredgecolor='black', linewidth=3, color='#F2C5C6', label='DE
Cats', linestyle='-')
plt.plot(overseas_df['Year'], overseas_df['DE_Dog'], marker='H',
markersize=10, markeredgecolor='black', linewidth=3, color='#a1d5b9', label='DE Dogs',
linestyle='-')
plt.title('DE Cats and Dogs (2019-2023)')
plt.xlabel('Year')
plt.ylabel('Number')
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

# 创建输出文件夹
desktop_path = os.path.join(os.path.expanduser("~"), "Desktop")
preprocess_folder = os.path.join(desktop_path, "pet_industry_preprocess_" +
datetime.now().strftime("%Y%m%d_%H%M%S"))
os.makedirs(preprocess_folder, exist_ok=True)

def check_data_quality(df, dataset_name):
    """
    检查数据质量并生成报告
    """
    report = f"\n{dataset_name} 数据质量检查报告:\n"
    report += "="*50 + "\n"

    # 1. 基础信息
    report += f"\n1. 基础信息:\n"
    report += f"数据维度: {df.shape}\n"
    report += f"列名: {list(df.columns)}\n"

    # 2. 重复值检查
    duplicates = df.duplicated().sum()
    report += f"\n2. 重复值检查:\n"
    report += f"重复行数: {duplicates}\n"

    # 3. 缺失值检查
    missing = df.isnull().sum()
    report += f"\n3. 缺失值检查:\n"
    report += f"每列缺失值数量:\n{missing}\n"

    # 4. 异常值检查 (使用Z-score方法)
    report += f"\n4. 异常值检查:\n"
    numeric_cols = df.select_dtypes(include=[np.number]).columns

    outliers_dict = {}
    for col in numeric_cols:
        z_scores = np.abs(stats.zscore(df[col]))
```

```

    outliers = np.where(z_scores > 3)[0]
    outliers_dict[col] = list(zip(outliers, df[col].iloc[outliers]))

    report += f"\n{col}列异常值:\n"
    if len(outliers) > 0:
        report += f"发现{len(outliers)}个潜在异常值:\n"
        for idx, value in outliers_dict[col]:
            report += f"索引{idx}: {value}\n"
    else:
        report += "未发现明显异常值\n"

    return report, outliers_dict

# 1. 中国宠物数据预处理
china_pets = {
    'Year': [2019, 2020, 2021, 2022, 2023],
    'Cat': [4412, 4862, 5806, 6536, 6980],
    'Dog': [5503, 5222, 5429, 5119, 5175]
}
china_df = pd.DataFrame(china_pets)
china_report, china_outliers = check_data_quality(china_df, "中国宠物数据")

# 2. 海外宠物数据预处理
overseas_data = {
    'Year': [2019, 2020, 2021, 2022, 2023],
    'US_Cat': [9420, 6500, 9420, 7380, 7380],
    'US_Dog': [8970, 8500, 8970, 8970, 8010],
    'FR_Cat': [1300, 1490, 1510, 1490, 1660],
    'FR_Dog': [740, 775, 750, 760, 990],
    'DE_Cat': [1470, 1570, 1670, 1520, 1570],
    'DE_Dog': [1010, 1070, 1030, 1060, 1050]
}
overseas_df = pd.DataFrame(overseas_data)
overseas_report, overseas_outliers = check_data_quality(overseas_df, "海外宠物数据")

# 3. 产业数据预处理
industry_data = {
    'Year': [2019, 2020, 2021, 2022, 2023],
    'Production': [440.7, 727.3, 1554, 1508, 2793],
    'Export': [154.1/7, 9.8, 12.2, 24.7, 39.6]
}
industry_df = pd.DataFrame(industry_data)
industry_report, industry_outliers = check_data_quality(industry_df, "产业数据")

import pandas as pd
import numpy as np

# 创建数据框
data = {
    'Year': [2019, 2020, 2021, 2022, 2023],
    'Cat': [4412, 4862, 5806, 6536, 6980],
    'Dog': [5503, 5222, 5429, 5119, 5175],
    'Production': [440.7, 727.3, 1554, 1508, 2793],
    'Pet_Population': [99.8, 108.5, 115.4, 122.6, 130.2],
    'Market_Size': [33.2, 35.6, 38.9, 42.1, 45.5],
    'Population': [6.88e8, 6.9e8, 6.91e8, 6.92e8, 6.91e8],
    'Population_Growth': [0.354741, 0.238041, 0.089252, -0.0131, -0.10379],
    'GDP': [10143.86, 10408.72, 12617.51, 12662.58, 12614.06]
}

df = pd.DataFrame(data)

# 特征工程

```

```
def feature_engineering(df):
    """
    进行特征工程
    """
    # 1. 计算年度增长率
    df['Cat_Growth'] = df['Cat'].pct_change() * 100
    df['Dog_Growth'] = df['Dog'].pct_change() * 100
    df['Production_Growth'] = df['Production'].pct_change() * 100
    df['Market_Size_Growth'] = df['Market_Size'].pct_change() * 100

    # 2. 计算市场结构特征
    df['Total_Pets'] = df['Cat'] + df['Dog']
    df['Cat_Ratio'] = df['Cat'] / df['Total_Pets'] * 100
    df['Dog_Ratio'] = df['Dog'] / df['Total_Pets'] * 100

    # 3. 计算人均指标
    df['GDP_per_capita'] = df['GDP'] / (df['Population'] / 1e8)
    df['Pets_per_capita'] = df['Total_Pets'] / (df['Population'] / 1e8)

    # 4. 计算市场效率指标
    df['Production_per_pet'] = df['Production'] / df['Total_Pets']
    df['Market_Size_per_pet'] = df['Market_Size'] / df['Total_Pets']

    # 5. 计算滚动平均（用于平滑趋势）
    df['Cat_MA'] = df['Cat'].rolling(window=2, min_periods=1).mean()
    df['Dog_MA'] = df['Dog'].rolling(window=2, min_periods=1).mean()

    # 6. 计算市场发展指数（综合指标）
    # 将各指标标准化后加权
    df['Market_Development_Index'] = (
        (df['Market_Size'] / df['Market_Size'].mean()) * 0.4 +
        (df['Production'] / df['Production'].mean()) * 0.3 +
        (df['Total_Pets'] / df['Total_Pets'].mean()) * 0.3
    )

    return df

# 应用特征工程
df_processed = feature_engineering(df)

# 显示处理后的数据基本信息
print("\n数据基本信息:")
print(df_processed.info())

# 显示新特征的描述性统计
print("\n新特征描述性统计:")
print(df_processed.describe())

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

# 1.1 宠物数量趋势分析
def analyze_pet_trends(df):
    plt.figure(figsize=(6, 4))
    plt.plot(df['Year'], df['Cat'], marker='o', markersize=10, markeredgewidth=3,
             color='#BFDEE5', label='猫咪数量')
    plt.plot(df['Year'], df['Dog'], marker='s', markersize=10, markeredgewidth=3,
             color='#e1abbc', label='狗狗数量')
    plt.title('2019-2023年中国宠物猫狗数量变化趋势')
```

```

plt.xlabel('年份')
plt.ylabel('数量(万只)')
plt.legend()
plt.show()

# 计算年均增长率
cat_cagr = ((df['Cat'].iloc[-1] / df['Cat'].iloc[0]) ** (1/4) - 1) * 100
dog_cagr = ((df['Dog'].iloc[-1] / df['Dog'].iloc[0]) ** (1/4) - 1) * 100

print(f'猫咪年均复合增长率: {cat_cagr:.2f}%')
print(f'狗狗年均复合增长率: {dog_cagr:.2f}%')

# 1.2 市场结构分析
def analyze_market_structure(df):
    colors = ['#BFDEE5', '#e1abbc']
    plt.figure(figsize=(6, 4))
    plt.stackplot(df['Year'],
                  [df['Cat_Ratio'], df['Dog_Ratio']],
                  labels=['猫咪占比', '狗狗占比'], colors=colors)
    plt.title('宠物市场结构变化')
    plt.xlabel('年份')
    plt.ylabel('占比(%)')
    plt.legend(loc='upper left')
    plt.show()

# 1.3 市场规模分析
def analyze_market_size(df):
    plt.figure(figsize=(6, 4))
    plt.plot(df['Year'], df['Production'], marker='o', markersize=10, markeredgcolor='black',
             linewidth=3, color='#a1d5b9', label='产值(亿元)')
    plt.title('宠物行业市场规模变化')
    plt.xlabel('年份')
    plt.ylabel('规模')
    plt.legend()
    plt.show()

# 执行分析
analyze_pet_trends(df_processed)
analyze_market_structure(df_processed)
analyze_market_size(df_processed)

# 生成分析报告
report = """
中国宠物行业2019-2023年发展分析报告

1. 宠物数量变化
- 猫咪数量持续增长，年均复合增长率为{:.2f}%
- 狗狗数量相对稳定，年均复合增长率为{:.2f}%
- 2021年是转折点，猫咪数量首次超过狗狗

2. 市场结构变化
- 猫咪市场份额持续提升
- 狗狗市场份额略有下降
- 市场结构更趋多元化

3. 产业规模发展
- 产值快速增长，从{:.1f}亿元增长到{:.1f}亿元
- 市场规模稳步扩大，从{:.1f}亿美元增长到{:.1f}亿美元
- 单位宠物产值显著提升
"""
format(
    ((df_processed['Cat'].iloc[-1]/df_processed['Cat'].iloc[0])**((1/4)-1)*100,

```

```

((df_processed['Dog'].iloc[-1]/df_processed['Dog'].iloc[0])**((1/4)-1)*100,
df_processed['Production'].iloc[0],
df_processed['Production'].iloc[-1],
df_processed['Market_Size'].iloc[0],
df_processed['Market_Size'].iloc[-1]
)
import numpy as np
import pandas as pd
from sklearn.model_selection import TimeSeriesSplit
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.linear_model import LinearRegression, BayesianRidge
import statsmodels.api as sm
from statsmodels.tsa.holtwinters import ExponentialSmoothing, SimpleExpSmoothing
from statsmodels.tsa.statespace.sarimax import SARIMAX
import pmdarima as pm
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
import os
import warnings
warnings.filterwarnings('ignore')

# 设置中文显示
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

def create_result_folder():
    """创建结果文件夹"""
    desktop_path = os.path.join(os.path.expanduser("~"), "Desktop")
    result_folder = os.path.join(desktop_path, "pet_industry_analysis_" +
datetime.now().strftime("%Y%m%d_%H%M%S"))
    os.makedirs(result_folder, exist_ok=True)
    return result_folder

def calculate_moving_average(data, window=3):
    """计算移动平均"""
    return np.convolve(data, np.ones(window)/window, mode='valid')

def multi_model_comparison(df, target_col):
    """
    比较多个预测模型的性能
    """
    results = {}
    predictions = {}

    # 准备数据
    data = df[target_col].values
    n = len(data)
    train = data[:-1]
    test = data[-1:]

    # 模型1: SARIMA
    try:
        sarima_model = pm.auto_arima(train, seasonal=False, stepwise=True,
suppress_warnings=True)
        sarima_pred = sarima_model.predict(n_periods=1)
        predictions['SARIMA'] = sarima_pred
        results['SARIMA'] = {
            'MSE': mean_squared_error(test, sarima_pred),
            'RMSE': np.sqrt(mean_squared_error(test, sarima_pred)),
            'MAE': mean_absolute_error(test, sarima_pred),
            'MAPE': np.mean(np.abs((test - sarima_pred) / test)) * 100,

```

```

        'Predicted_Value': sarima_pred[0]
    }
    print(f'SARIMA model successful for {target_col}')
except Exception as e:
    print(f'SARIMA model failed for {target_col}: {str(e)}')

# 模型2: Holt-Winters
try:
    hw_model = ExponentialSmoothing(train, trend='add', seasonal=None)
    hw_fitted = hw_model.fit()
    hw_pred = hw_fitted.forecast(1)
    predictions['Holt-Winters'] = hw_pred
    results['Holt-Winters'] = {
        'MSE': mean_squared_error(test, hw_pred),
        'RMSE': np.sqrt(mean_squared_error(test, hw_pred)),
        'MAE': mean_absolute_error(test, hw_pred),
        'MAPE': np.mean(np.abs((test - hw_pred) / test)) * 100,
        'Predicted_Value': hw_pred[0]
    }
    print(f'Holt-Winters model successful for {target_col}')
except Exception as e:
    print(f'Holt-Winters model failed for {target_col}: {str(e)}')

# 模型3: 简单线性回归
try:
    X = np.arange(len(train)).reshape(-1, 1)
    lr = LinearRegression()
    lr.fit(X, train)
    lr_pred = lr.predict(np.array([[len(train)]]))
    predictions['Linear'] = lr_pred
    results['Linear'] = {
        'MSE': mean_squared_error(test, lr_pred),
        'RMSE': np.sqrt(mean_squared_error(test, lr_pred)),
        'MAE': mean_absolute_error(test, lr_pred),
        'MAPE': np.mean(np.abs((test - lr_pred) / test)) * 100,
        'Predicted_Value': lr_pred[0]
    }
    print(f'Linear model successful for {target_col}')
except Exception as e:
    print(f'Linear model failed for {target_col}: {str(e)}')

# 新增模型4: 简单移动平均
try:
    ma_values = calculate_moving_average(train)
    ma_pred = np.array([ma_values[-1]]) # 使用最后一个移动平均值作为预测
    predictions['MovingAverage'] = ma_pred
    results['MovingAverage'] = {
        'MSE': mean_squared_error(test, ma_pred),
        'RMSE': np.sqrt(mean_squared_error(test, ma_pred)),
        'MAE': mean_absolute_error(test, ma_pred),
        'MAPE': np.mean(np.abs((test - ma_pred) / test)) * 100,
        'Predicted_Value': ma_pred[0]
    }
    print(f'Moving Average model successful for {target_col}')
except Exception as e:
    print(f'Moving Average model failed for {target_col}: {str(e)}')

# 新增模型5: 简单指数平滑
try:
    ses_model = SimpleExpSmoothing(train).fit()
    ses_pred = ses_model.forecast(1)
    predictions['ExpSmoothing'] = ses_pred
    results['ExpSmoothing'] = {
        'MSE': mean_squared_error(test, ses_pred),
        'RMSE': np.sqrt(mean_squared_error(test, ses_pred)),

```



```

        'MAE': mean_absolute_error(test, ses_pred),
        'MAPE': np.mean(np.abs((test - ses_pred) / test)) * 100,
        'Predicted_Value': ses_pred[0]
    }
    print(f'Simple Exponential Smoothing model successful for {target_col}')
except Exception as e:
    print(f'Simple Exponential Smoothing model failed for {target_col}: {str(e)}')

# 新增模型6: 多项式回归
try:
    X = np.arange(len(train)).reshape(-1, 1)
    poly = PolynomialFeatures(degree=2)
    X_poly = poly.fit_transform(X)
    poly_reg = LinearRegression()
    poly_reg.fit(X_poly, train)
    X_pred = poly.transform(np.array([[len(train)]]))
    poly_pred = poly_reg.predict(X_pred)
    predictions['Polynomial'] = poly_pred
    results['Polynomial'] = {
        'MSE': mean_squared_error(test, poly_pred),
        'RMSE': np.sqrt(mean_squared_error(test, poly_pred)),
        'MAE': mean_absolute_error(test, poly_pred),
        'MAPE': np.mean(np.abs((test - poly_pred) / test)) * 100,
        'Predicted_Value': poly_pred[0]
    }
    print(f'Polynomial Regression model successful for {target_col}')
except Exception as e:
    print(f'Polynomial Regression model failed for {target_col}: {str(e)}')

# 新增模型7: 贝叶斯岭回归
try:
    X = np.arange(len(train)).reshape(-1, 1)
    br = BayesianRidge()
    br.fit(X, train)
    br_pred = br.predict(np.array([[len(train)]]))
    predictions['BayesianRidge'] = br_pred
    results['BayesianRidge'] = {
        'MSE': mean_squared_error(test, br_pred),
        'RMSE': np.sqrt(mean_squared_error(test, br_pred)),
        'MAE': mean_absolute_error(test, br_pred),
        'MAPE': np.mean(np.abs((test - br_pred) / test)) * 100,
        'Predicted_Value': br_pred[0]
    }
    print(f'Bayesian Ridge model successful for {target_col}')
except Exception as e:
    print(f'Bayesian Ridge model failed for {target_col}: {str(e)}')

if not predictions:
    raise ValueError(f'No models were successfully fitted for {target_col}')

results_df = pd.DataFrame(results).T
print(f'\nResults for {target_col}:')
print(results_df)

# 选择最佳模型（基于MAPE）
best_model = results_df['MAPE'].idxmin()
print(f'\nBest model for {target_col}: {best_model}')

return results_df, predictions, best_model

def residual_analysis(actual, predicted, model_name):
    """
    进行残差分析
    """

```

```
residuals = actual - predicted

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(15, 12))

# 残差散点图
sns.scatterplot(x=predicted, y=residuals, ax=ax1)
ax1.axhline(y=0, color='r', linestyle='--')
ax1.set_title(f'{model_name} 残差散点图')
ax1.set_xlabel('预测值')
ax1.set_ylabel('残差')

# 残差直方图
sns.histplot(residuals, ax=ax2, kde=True)
ax2.set_title('残差分布')

# Q-Q图
stats.probplot(residuals, dist="norm", plot=ax3)
ax3.set_title('Q-Q图')

# 残差自相关图
sm.graphics.tsa.plot_acf(residuals, ax=ax4)
ax4.set_title('残差自相关图')

plt.tight_layout()
return fig

def bootstrap_prediction_intervals(data, n_bootstrap=1000, confidence=0.95):
    """
    使用bootstrap方法进行区间估计
    """
    predictions = np.zeros((n_bootstrap, 3))

    for i in range(n_bootstrap):
        indices = np.random.randint(0, len(data), size=len(data))
        sample = data[indices]

        try:
            model = ExponentialSmoothing(sample, trend='add', seasonal=None)
            fitted = model.fit()
            pred = fitted.forecast(3)
            predictions[i] = pred
        except:
            # 如果指数平滑失败，使用简单的移动平均
            ma_values = calculate_moving_average(sample)
            pred = np.array([ma_values[-1]] * 3)
            predictions[i] = pred

    # 计算预测区间
    lower = np.percentile(predictions, ((1 - confidence) / 2) * 100, axis=0)
    upper = np.percentile(predictions, (1 - (1 - confidence) / 2) * 100, axis=0)
    mean = np.mean(predictions, axis=0)

    return pd.DataFrame({
        'mean': mean,
        'lower': lower,
        'upper': upper
    }, index=['2024', '2025', '2026'])

def sensitivity_analysis(df, target_col, best_model='BayesianRidge'):
    """
    对贝叶斯岭回归模型进行敏感性分析
    """
    results = {}
```

```
# 1. 参数敏感性分析
param_sensitivity = analyze_parameter_sensitivity(df[target_col])

# 2. 数据扰动分析
perturbation_sensitivity = analyze_data_perturbation(df[target_col])

# 3. 时间窗口敏感性
window_sensitivity = analyze_time_window(df[target_col])

return {
    '参数敏感性': param_sensitivity,
    '数据扰动敏感性': perturbation_sensitivity,
    '时间窗口敏感性': window_sensitivity
}

def analyze_parameter_sensitivity(data):
    """
    分析模型对不同超参数的敏感性
    """
    X = np.arange(len(data)).reshape(-1, 1)
    y = data.values

    # 测试不同的alpha值
    alpha_values = [1e-6, 1e-4, 1e-2, 1, 10, 100]
    alpha_results = {}

    for alpha in alpha_values:
        model = BayesianRidge(alpha_1=alpha, alpha_2=alpha, lambda_1=alpha, lambda_2=alpha)
        model.fit(X[:-1], y[:-1])
        pred = model.predict(X[-1:])
        mape = np.mean(np.abs((y[-1] - pred) / y[-1])) * 100
        alpha_results[f'alpha_{alpha}'] = {
            '预测值': pred[0],
            'MAPE': mape
        }

    return pd.DataFrame(alpha_results).T

def analyze_data_perturbation(data, n_iterations=100, perturbation_range=0.05):
    """
    分析模型对数据扰动的敏感性
    """
    X = np.arange(len(data)).reshape(-1, 1)
    y = data.values
    results = []

    model = BayesianRidge()

    for _ in range(n_iterations):
        # 添加随机扰动
        noise = np.random.normal(0, perturbation_range, size=len(y))
        y_perturbed = y * (1 + noise)

        # 训练模型
        model.fit(X[:-1], y_perturbed[:-1])
        pred = model.predict(X[-1:])
        mape = np.mean(np.abs((y[-1] - pred) / y[-1])) * 100
        results.append({
            '预测值': pred[0],
            'MAPE': mape
        })

    return pd.DataFrame(results).describe()
```

```
def analyze_time_window(data, min_window=5):
    """
    分析模型对时间窗口长度的敏感性
    """
    X = np.arange(len(data)).reshape(-1, 1)
    y = data.values
    window_results = {}

    for window in range(min_window, len(data)):
        model = BayesianRidge()
        model.fit(X[len(data)-window-1:-1], y[len(data)-window-1:-1])
        pred = model.predict(X[-1:])
        mape = np.mean(np.abs((y[-1] - pred) / y[-1])) * 100
        window_results[f'window_{window}'] = {
            '预测值': pred[0],
            'MAPE': mape
        }

    return pd.DataFrame(window_results).T

def run_full_analysis(df_processed):
    """
    执行完整分析
    """
    # 创建结果文件夹
    result_folder = create_result_folder()
    forecast_report = {}

    try:
        # 1. 多模型比较
        print("\n执行多模型比较...")
        cat_results, cat_predictions, cat_best_model = multi_model_comparison(df_processed, 'Cat')
        dog_results, dog_predictions, dog_best_model = multi_model_comparison(df_processed,
'Dog')

        forecast_report['模型比较'] = {
            '猫咪': cat_results,
            '狗狗': dog_results
        }

        # 2. 残差分析
        print("\n执行残差分析...")
        cat_residual_fig = residual_analysis(
            df_processed['Cat'].values,
            np.append(df_processed['Cat'].values[:-1], cat_predictions[cat_best_model]),
            f'猫咪 {cat_best_model} 模型'
        )

        dog_residual_fig = residual_analysis(
            df_processed['Dog'].values,
            np.append(df_processed['Dog'].values[:-1], dog_predictions[dog_best_model]),
            f'狗狗 {dog_best_model} 模型'
        )

        # 保存残差分析图
        print(f'保存残差分析图到: {result_folder}')
        cat_residual_fig.savefig(os.path.join(result_folder, '猫咪残差分析.png'))
        plt.close(cat_residual_fig)
        dog_residual_fig.savefig(os.path.join(result_folder, '狗狗残差分析.png'))
        plt.close(dog_residual_fig)
```

```

# 3. 预测未来三年
future_predictions = {
    '猫咪': {'2024': None, '2025': None, '2026': None},
    '狗狗': {'2024': None, '2025': None, '2026': None}
}

# 使用最佳模型进行预测
for animal, (best_model, data) in [('猫咪', (cat_best_model, df_processed['Cat'])),
                                   ('狗狗', (dog_best_model, df_processed['Dog']))]:
    if best_model == 'MovingAverage':
        ma_values = calculate_moving_average(data.values)
        future = np.array([ma_values[-1]] * 3)
    elif best_model == 'ExpSmoothing':
        model = SimpleExpSmoothing(data.values).fit()
        future = model.forecast(3)
    elif best_model == 'Polynomial':
        X = np.arange(len(data)).reshape(-1, 1)
        poly = PolynomialFeatures(degree=2)
        X_poly = poly.fit_transform(X)
        poly_reg = LinearRegression()
        poly_reg.fit(X_poly, data.values)
        future_X = np.array([[len(data)], [len(data)+1], [len(data)+2]])
        X_future_poly = poly.transform(future_X)
        future = poly_reg.predict(X_future_poly)
    else: # 默认使用Holt-Winters
        model = ExponentialSmoothing(data.values, trend='add', seasonal=None)
        fitted = model.fit()
        future = fitted.forecast(3)

    future_predictions[animal] = {
        '2024': future[0],
        '2025': future[1],
        '2026': future[2]
    }

forecast_report['未来预测'] = future_predictions

# 4. 计算预测区间
print("\n计算预测区间...")
cat_intervals = bootstrap_prediction_intervals(df_processed['Cat'].values)
dog_intervals = bootstrap_prediction_intervals(df_processed['Dog'].values)

forecast_report['预测区间'] = {
    '猫咪': cat_intervals,
    '狗狗': dog_intervals
}

print("\n执行敏感性分析...")
cat_sensitivity = sensitivity_analysis(df_processed, 'Cat')
dog_sensitivity = sensitivity_analysis(df_processed, 'Dog')

forecast_report['敏感性分析'] = {
    '猫咪': cat_sensitivity,
    '狗狗': dog_sensitivity
}

except Exception as e:
    print(f"Error in analysis: {str(e)}")
    raise

return forecast_report, result_folder

```

```
# 执行分析
try:
    forecast_report, result_folder = run_full_analysis(df_processed)

    # 打印主要结果
    print("\n=== 分析结果汇总 ===")
    print(f"\n结果保存在: {result_folder}")
    print("\n1. 模型比较结果（猫咪）:")
    print(forecast_report['模型比较']['猫咪'])
    print("\n2. 模型比较结果（狗狗）:")
    print(forecast_report['模型比较']['狗狗'])
    print("\n3. 未来预测结果:")
    print("\n猫咪未来预测:")
    print(pd.DataFrame(forecast_report['未来预测']['猫咪'], index=['预测值']).T)
    print("\n狗狗未来预测:")
    print(pd.DataFrame(forecast_report['未来预测']['狗狗'], index=['预测值']).T)
    print("\n4. 预测区间:")
    print("\n猫咪预测区间:")
    print(forecast_report['预测区间']['猫咪'])
    print("\n狗狗预测区间:")
    print(forecast_report['预测区间']['狗狗'])

    # 保存结果
    save_results(forecast_report, result_folder)
    print("\n分析结果已保存到文件")

except Exception as e:
    print(f'Analysis failed: {str(e)}')
    print(f'Error occurred at: {str(e.__traceback__.tb_lineno)}')
```

No: 2	The Second question
	<pre> import pandas as pd  # 创建中国数据DataFrame china_data = pd.DataFrame({     '国家': ['中国']*5,     '年份': [2019, 2020, 2021, 2022, 2023],     '猫(万)': [4412, 4862, 5806, 6536, 6980],     '狗(万)': [5503, 5222, 5429, 5119, 5175],     '宠物数量(百万)': [99.8, 108.5, 115.4, 122.6, 130.2],     '宠物市场规模(亿美元)': [33.2, 35.6, 38.9, 42.1, 45.5],     '宠物食品开支(亿美元)': [15.1, 16.2, 17.5, 18.9, 20.3],     '人均GDP(美元)': [10143.86, 10408.72, 12617.51, 12662.58, 12614.06],     '宠物家庭渗透率': [0.18, 0.20, 0.20, 0.20, 0.22] })  # 创建美国数据DataFrame us_data = pd.DataFrame({     '国家': ['美国']*5,     '年份': [2019, 2020, 2021, 2022, 2023],     '猫(万)': [9420, 6500, 9420, 7380, 7380],     '狗(万)': [8970, 8500, 8970, 8970, 8010],     '宠物数量(百万)': [89.7, 91.8, 94.2, 96.7, 99.2],     '宠物市场规模(亿美元)': [95.7, 98.3, 101.2, 104.5, 108.0],     '宠物食品开支(亿美元)': [36.9, 38.1, 39.4, 40.8, 42.3],     '人均GDP(美元)': [65548.07, 64317.40, 71055.88, 77246.67, 81695.19],     '宠物家庭渗透率': [0.67, 0.70, 0.70, 0.70, 0.66] })  # 创建法国数据DataFrame france_data = pd.DataFrame({     '国家': ['法国']*5,     '年份': [2019, 2020, 2021, 2022, 2023],     '猫(万)': [1300, 1490, 1510, 1490, 1660],     '狗(万)': [740, 775, 750, 760, 990],     '宠物数量(百万)': [27.3, 28.0, 28.8, 29.7, 30.6],     '宠物市场规模(亿美元)': [5.7, 6.0, 6.3, 6.7, 7.1],     '宠物食品开支(亿美元)': [2.6, 2.8, 3.0, 3.2, 3.4],     '人均GDP(美元)': [40494.90, 39179.74, 43671.31, 40886.25, 44460.82],     '宠物家庭渗透率': [0.52, 0.52, 0.52, 0.52, 0.52] })  # 创建德国数据DataFrame germany_data = pd.DataFrame({     '国家': ['德国']*5,     '年份': [2019, 2020, 2021, 2022, 2023],     '猫(万)': [1470, 1570, 1670, 1520, 1570],     '狗(万)': [1010, 1070, 1030, 1060, 1050],     '宠物数量(百万)': [34.3, 35.0, 36.0, 37.2, 38.5],     '宠物市场规模(亿美元)': [6.6, 6.8, 7.2, 7.6, 8.0],     '宠物食品开支(亿美元)': [3.0, 3.2, 3.4, 3.6, 3.8], </pre>

```
'人均GDP(美元)': [46805.14, 46749.48, 51426.75, 48717.99, 52745.76],
'宠物家庭渗透率': [0.57, 0.57, 0.57, 0.57, 0.57]
})

# 合并所有数据
df = pd.concat([china_data, us_data, france_data, germany_data], ignore_index=True)
# 定义预测函数
def predict_food_value(pets_count):
    return 0.8796 * pets_count - 8279.1465

# 打印确认数据已正确合并
print("数据框架的形状:", df.shape)
print("\n数据框架的前几行:")
print(df.head())

from statsmodels.stats.outliers_influence import variance_inflation_factor
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from datetime import datetime
import os

# 1. 创建扩展特征
def create_extended_features(df):
    """创建扩展特征集"""
    extended_df = df.copy()

    # 添加新特征
    extended_df['人均宠物数'] = (extended_df['猫(万)'] + extended_df['狗(万)']) /
extended_df['宠物数量(百万)']
    extended_df['人均市场规模'] = extended_df['宠物市场规模(亿美元)'] /
extended_df['宠物数量(百万)']
    extended_df['食品开支占比'] = extended_df['宠物食品开支(亿美元)'] /
extended_df['宠物市场规模(亿美元)']
    extended_df['人均食品开支'] = extended_df['宠物食品开支(亿美元)'] /
extended_df['宠物数量(百万)']
    extended_df['GDP渗透率'] = extended_df['宠物市场规模(亿美元)'] /
extended_df['人均GDP(美元)']
    extended_df['猫狗比例'] = extended_df['猫(万)'] / extended_df['狗(万)']
    extended_df['总宠物支出'] = extended_df['宠物市场规模(亿美元)'] *
extended_df['宠物家庭渗透率']
    extended_df['人均宠物GDP'] = extended_df['宠物市场规模(亿美元)'] /
extended_df['人均GDP(美元)']
    extended_df['食品人均支出'] = extended_df['宠物食品开支(亿美元)'] /
(extended_df['猫(万)'] + extended_df['狗(万)'])

    # 移除非数值列
    extended_df = extended_df.drop(['国家', '年份'], axis=1)

    return extended_df
```



```
# 2. VIF分析函数
def calculate_vif(X):
    """计算VIF值"""
    vif_data = pd.DataFrame()
    vif_data["Feature"] = X.columns
    vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in
range(X.shape[1])]
    return vif_data.sort_values('VIF', ascending=False)

# 3. 特征选择函数
def select_features_by_vif(df, target_col, threshold=5.0):
    """基于VIF阈值选择特征"""
    # 初始特征集
    features = df.columns.tolist()
    features.remove(target_col)

    print("初始特征集:", features)
    print("\n开始VIF筛选过程...")

    selected_features = features.copy()
    while True:
        if len(selected_features) < 2: # 至少需要两个特征
            break

        X = df[selected_features]
        vif = calculate_vif(X)

        print("\n当前VIF值:")
        print(vif)

        if vif['VIF'].max() <= threshold:
            break

        worst_feature = vif.iloc[0]['Feature']
        selected_features.remove(worst_feature)
        print(f"\n移除特征: {worst_feature} (VIF: {vif.iloc[0]['VIF']:.2f})")

    print("\n最终选择的特征:", selected_features)
    return selected_features

# 4. 主执行函数
def feature_selection_process(df, target_col='宠物食品开支(亿美元)',
vif_threshold=5.0):
    """完整的特征选择流程"""
    print("开始特征选择流程...")

    # 1. 创建扩展特征
    print("\n1. 创建扩展特征")
    extended_df = create_extended_features(df)
    print("扩展后的特征集:", extended_df.columns.tolist())

    # 2. VIF分析和特征选择
    print("\n2. 开始VIF分析")
    selected_features = select_features_by_vif(extended_df, target_col, vif_threshold)
```

```
# 3. 创建最终数据集
final_df = extended_df[selected_features + [target_col]]

# 4. 计算相关性
print("\n3. 计算特征与目标变量的相关性")
correlations = final_df.corr()[target_col].sort_values(ascending=False)
print("\n特征与目标变量的相关性:")
print(correlations)

# 5. 可视化相关性矩阵
plt.figure(figsize=(6, 4))
sns.heatmap(final_df.corr(), annot=True, fmt='.2f', cmap=cmap, center=0)
plt.title('Final Features Correlation Matrix')
plt.tight_layout()
plt.show()

return final_df, selected_features

# 5. 执行特征选择
final_df, selected_features = feature_selection_process(df, vif_threshold=5.0)

# 6. 输出结果
print("\n最终选择的特征集:")
for i, feature in enumerate(selected_features, 1):
    print(f'{i}. {feature}')

from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
from scipy import stats

# 1. 定义模型集合
def get_models():
    models = {
        'Linear Regression': LinearRegression(),
        'Ridge': Ridge(alpha=1.0),
        'Lasso': Lasso(alpha=1.0),
        'ElasticNet': ElasticNet(alpha=1.0, l1_ratio=0.5),
        'Random Forest': RandomForestRegressor(n_estimators=100, random_state=42),
        'Gradient Boosting': GradientBoostingRegressor(n_estimators=100,
random_state=42),
        'SVR': SVR(kernel='rbf')
    }
    return models

# 2. Bootstrap函数
def bootstrap_predictions(model, X, y, X_future, n_iterations=1000):
    predictions = np.zeros((n_iterations, len(X_future)))
    n_samples = len(X)

    for i in range(n_iterations):
        # Bootstrap采样
```

```
indices = np.random.randint(0, n_samples, n_samples)
X_boot = X.iloc[indices]
y_boot = y.iloc[indices]

# 训练模型并预测
model.fit(X_boot, y_boot)
predictions[i] = model.predict(X_future)

# 计算预测区间
lower = np.percentile(predictions, 2.5, axis=0)
upper = np.percentile(predictions, 97.5, axis=0)
mean = np.mean(predictions, axis=0)

return mean, lower, upper

# 3. 模型评估函数
def evaluate_models(models, X_train, X_test, y_train, y_test):
    results = {}

    for name, model in models.items():
        # 训练模型
        model.fit(X_train, y_train)

        # 预测
        y_pred = model.predict(X_test)

        # 计算评估指标
        mse = mean_squared_error(y_test, y_pred)
        r2 = r2_score(y_test, y_pred)

        results[name] = {
            'MSE': mse,
            'R2': r2,
            'Model': model
        }

    return results

# 4. 预测未来值
def predict_future_with_bootstrap(best_model, X, y, future_features):
    # 使用bootstrap进行预测
    mean_pred, lower_pred, upper_pred = bootstrap_predictions(
        best_model, X, y, future_features
    )

    return mean_pred, lower_pred, upper_pred

# 5. 执行预测流程
def run_prediction_analysis(df):
    # 准备特征
    features = [
        '宠物总数(万)',
        '人均GDP(美元)',
        '宠物家庭渗透率',
```

```
'宠物数量(百万)',
'宠物市场规模(亿美元)'
]

X = df[features]
y = df['宠物食品开支(亿美元)']

# 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# 获取并评估模型
models = get_models()
results = evaluate_models(models, X_train, X_test, y_train, y_test)

# 打印评估结果
print("\n模型评估结果:")
for name, result in results.items():
    print(f"\n{name}:")
    print(f'MSE: {result["MSE"]:.4f}')
    print(f'R2: {result["R2"]:.4f}')

# 选择最佳模型
best_model_name = max(results.items(), key=lambda x: x[1]["R2"])[0]
best_model = results[best_model_name]["Model"]

return best_model, features

# 6. 可视化结果
def plot_predictions_with_intervals(df, predictions, intervals, countries):
    plt.figure(figsize=(12, 6))
    colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728']

    for i, country in enumerate(countries):
        # 历史数据
        historical = df[df['国家'] == country]
        plt.plot(historical['年份'], historical['宠物食品开支(亿美元)'],
                 color=colors[i], marker='o', label=f'{country} (历史)')

        # 预测值和区间
        future_years = range(2024, 2027)
        plt.plot(future_years, predictions[country],
                 color=colors[i], linestyle='--', marker='s',
                 label=f'{country} (预测)')

        # 置信区间
        plt.fill_between(future_years,
                        intervals[country]['lower'],
                        intervals[country]['upper'],
                        color=colors[i], alpha=0.2)

    plt.title('宠物食品开支预测 (2024-2026)')
    plt.xlabel('年份')
```

```
plt.ylabel('食品开支 (亿美元)')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

# 7. 主执行流程
# 准备数据
best_model, features = run_prediction_analysis(df)

# 为每个国家进行预测
countries = df['国家'].unique()
predictions = {}
intervals = {}

for country in countries:
    country_data = df[df['国家'] == country]

    # 准备未来特征
    future_features = pd.DataFrame()
    for feature in features:
        # 使用简单线性回归预测特征值
        years = np.array(range(2019, 2024))
        future_years = np.array(range(2024, 2027))

        feature_model = LinearRegression()
        feature_model.fit(years.reshape(-1, 1), country_data[feature])
        future_values = feature_model.predict(future_years.reshape(-1, 1))

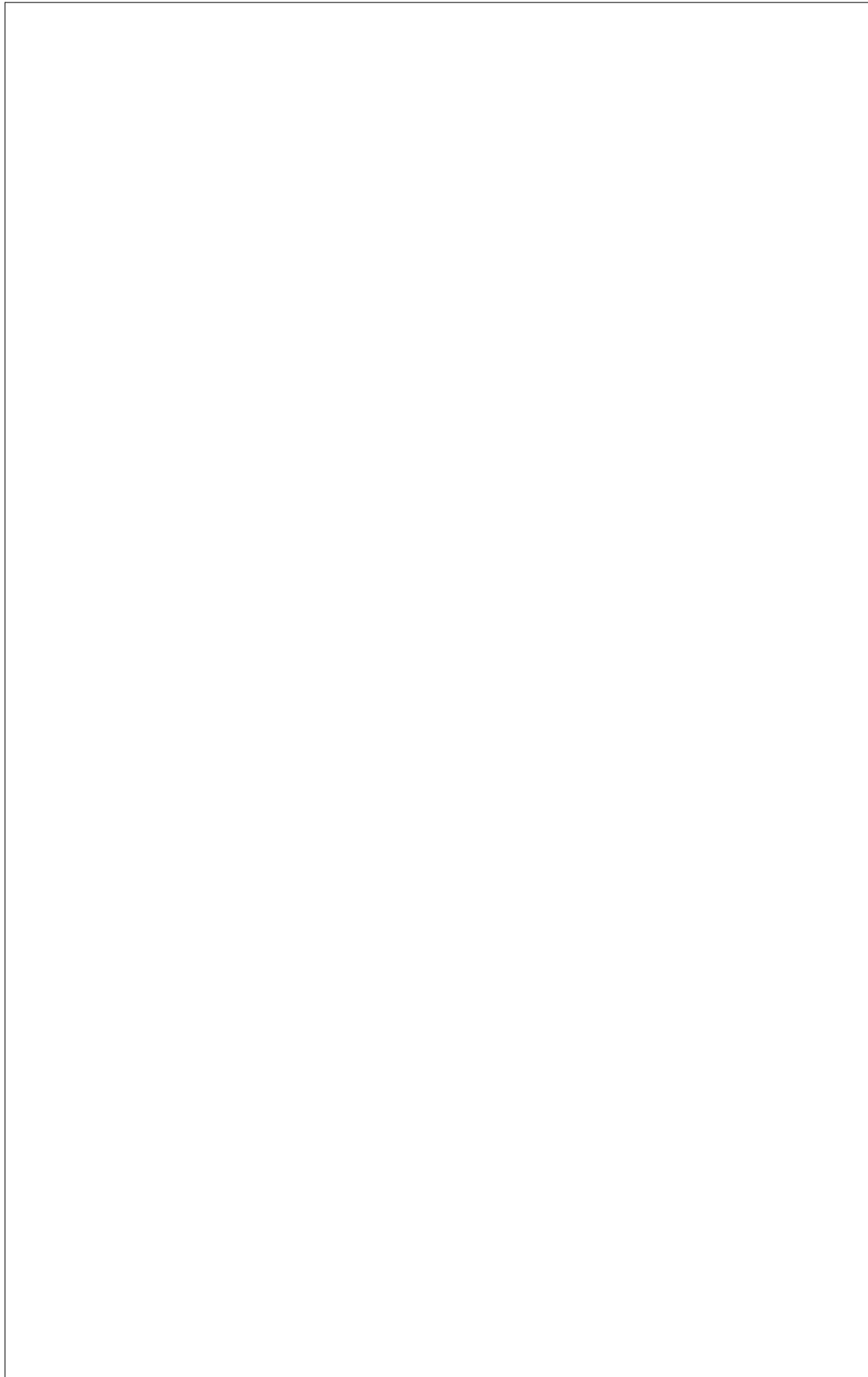
        if len(future_features) == 0:
            future_features = pd.DataFrame(index=range(len(future_years)))
        future_features[feature] = future_values

    # 使用bootstrap进行预测
    mean_pred, lower_pred, upper_pred = predict_future_with_bootstrap(
        best_model, df[features], df['宠物食品开支(亿美元)'], future_features
    )

    predictions[country] = mean_pred
    intervals[country] = {
        'lower': lower_pred,
        'upper': upper_pred
    }

# 8. 可视化结果
plot_predictions_with_intervals(df, predictions, intervals, countries)

# 9. 输出详细预测结果
print("\n预测结果 (单位: 亿美元):")
for country in countries:
    print(f"\n{country}:")
    for i, year in enumerate(range(2024, 2027)):
        print(f'{year}: {predictions[country][i]:.2f} " +
              f"[{intervals[country]['lower'][i]:.2f}, " +
              f"{intervals[country]['upper'][i]:.2f}]")
```



No: 3	The Third Question
	<pre> china_pet_data = {     'year': [2019, 2020, 2021, 2022, 2023],      # 宠物数量（万只）     'cats': [4412, 4862, 5806, 6536, 6980],     'dogs': [5503, 5222, 5429, 5119, 5175],     'total_pets': [99.8, 105.5, 115.4, 122.6, 130.2], # 单位：百万      # 市场数据     'market_scale': [33.2, 35.6, 38.9, 42.1, 45.5], # 宠物市场规模（亿美元）     'per_capita_spending': [15.1, 16.2, 17.5, 18.9, 20.3], # 人均开支（亿美元）     'gdp': [10143.86, 10408.72, 9617.51, 12662.58, 12614.06], # GDP（现价美元）     'penetration_rate': [0.18, 0.20, 0.24, 0.20, 0.22] # 宠物家庭渗透率 }  # 宠物食品生产和出口数据 production_export_data = {     'year': [2019, 2020, 2021, 2022, 2023],      # 生产总值（亿元人民币）     'production_value': [440.7, 727.3, 1554.0, 1508.0, 2793.0],      # 出口值     'export_value': [154.1/7.0, 9.8, 12.2, 24.7, 39.6] # 2019年原始数据为人民币，已 转换为美元 }  # 第二问预测的全球主要市场宠物食品开支（亿美元） global_forecast = {     'country': ['China', 'America', 'France', 'Germany'],     '2024': [21.52, 43.55, 3.47, 3.97],     '2025': [22.83, 44.89, 3.63, 4.15],     '2026': [24.13, 46.22, 3.79, 4.34] }  import pandas as pd import numpy as np from sklearn.linear_model import LinearRegression import matplotlib.pyplot as plt  # 1. 数据准备 years = np.array([2019, 2020, 2021, 2022, 2023]) total_pets = np.array([99.8, 105.5, 115.4, 122.6, 130.2]) market_scale = np.array([33.2, 35.6, 38.9, 42.1, 45.5]) per_capita_spending = np.array([15.1, 16.2, 17.5, 18.9, 20.3]) production_value = np.array([440.7, 727.3, 1554.0, 1508.0, 2793.0]) export_value = np.array([154.1/7.0, 9.8, 12.2, 24.7, 39.6])  # 2. 相关性分析 data = pd.DataFrame({     'total_pets': total_pets,     'market_scale': market_scale,     'per_capita_spending': per_capita_spending, </pre>

```

'production_value': production_value,
'export_value': export_value
})

correlation_matrix = data.corr()
print("相关性矩阵: ")
print(correlation_matrix['production_value'])

import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import matplotlib.pyplot as plt
from statsmodels.stats.outliers_influence import variance_inflation_factor

# 整合所有数据
data = pd.DataFrame({
    'year': [2019, 2020, 2021, 2022, 2023],

    # 附件3数据
    'production_value': [440.7, 727.3, 1554.0, 1508.0, 2793.0], # 亿元人民币
    'export_value': [154.1/7.0, 9.8, 12.2, 24.7, 39.6], # 亿美元

    # 您收集的数据
    'cats': [4412, 4862, 5806, 6536, 6980], # 万只
    'dogs': [5503, 5222, 5429, 5119, 5175], # 万只
    'total_pets': [99.8, 105.5, 115.4, 122.6, 130.2], # 百万只
    'market_scale': [33.2, 35.6, 38.9, 42.1, 45.5], # 亿美元
    'per_capita_spending': [15.1, 16.2, 17.5, 18.9, 20.3], # 亿美元
    'penetration_rate': [0.18, 0.20, 0.24, 0.20, 0.22]
})

# 1. 产业规模增长率
growth_rate = (data['production_value'].pct_change() * 100)

# 2. 出口依存度 - 有明确的计算方法
export_dependency = (data['export_value'] * 7 / data['production_value']) * 100

# 3. 产业产出效率 - 基于实际产值数据
output_efficiency = data['production_value'] / data['total_pets']

# 可视化这些基础指标
plt.figure(figsize=(15, 5))

# 1. 产业规模增长趋势
plt.subplot(1, 3, 1)
plt.plot(data['year'][1:], growth_rate[1:], marker='s',
markersize=10, markeredgecolor='black', linewidth=3, color='#F2C5C6')
plt.title('Industry Growth Rate (%)', fontsize=12)
plt.grid(True)

# 2. 出口依存度趋势
plt.subplot(1, 3, 2)
plt.plot(data['year'], export_dependency, marker='s',

```



```
markersize=10,markeredgecolor='black', linewidth=3, color='#F2C5C6')
plt.title('Export Dependency (%)', fontsize=12)
plt.grid(True)

# 3. 产出效率趋势
plt.subplot(1, 3, 3)
plt.plot(data['year'], output_efficiency, marker='s',
markersize=10,markeredgecolor='black', linewidth=3, color='#F2C5C6')
plt.title('Output Efficiency', fontsize=12)
plt.grid(True)

plt.tight_layout()
plt.show()

# 输出分析结果
print("\nChina's Pet Food Industry Analysis (Based on Official Data):")
print("="*80)

# 1. 产业规模分析
print("\n1. Industry Scale Analysis:")
print(f"- Average Annual Growth Rate: {growth_rate.mean():.1f}%")
print(f"- Growth Rate Volatility: {growth_rate.std():.1f}%")

# 2. 出口分析
print("\n2. Export Analysis:")
print(f"- Current Export Dependency: {export_dependency.iloc[-1]:.1f}%")
print(f"- Export Dependency Change: {export_dependency.iloc[-1] -
export_dependency.iloc[0]:.1f} percentage points")

# 3. 效率分析
efficiency_growth = ((output_efficiency.iloc[-1]/output_efficiency.iloc[0]) - 1) * 100
print("\n3. Efficiency Analysis:")
print(f"- Output Efficiency Growth: {efficiency_growth:.1f}%")

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from sklearn.linear_model import BayesianRidge, LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from statsmodels.tsa.api import SimpleExpSmoothing
import warnings
warnings.filterwarnings('ignore')

# 准备数据
y = data['production_value']
X = data[['export_value', 'market_scale']]

# 定义时间序列模型和预测函数
class TimeSeriesModels:
    def __init__(self, data, future_steps=3):
        self.data = data
        self.future_steps = future_steps

    def holt_winters_forecast(self):
```

```
        model = ExponentialSmoothing(self.data,
                                      seasonal_periods=2,
                                      trend='add',
                                      seasonal='add',
                                      initialization_method='estimated')
        results = model.fit()
        forecast = results.forecast(self.future_steps)
        return forecast

    def exp_smoothing_forecast(self):
        model = SimpleExpSmoothing(self.data)
        results = model.fit()
        forecast = results.forecast(self.future_steps)
        return forecast

    def moving_average(self, window=2):
        ma = self.data.rolling(window=window).mean()
        last_ma = ma.iloc[-1]
        trend = (ma.iloc[-1] - ma.iloc[-2]) if len(ma) > 1 else 0
        forecast = np.array([last_ma + i * trend for i in range(self.future_steps)])
        return forecast

# 定义回归模型
class RegressionModels:
    def __init__(self, X, y):
        self.X = X
        self.y = y

    def linear_regression(self):
        model = LinearRegression()
        model.fit(self.X, self.y)
        return model

    def polynomial_regression(self, degree=2):
        poly = PolynomialFeatures(degree=degree)
        X_poly = poly.fit_transform(self.X)
        model = BayesianRidge(max_iter=300)
        model.fit(X_poly, self.y)
        return model, poly

    def bayesian_ridge(self):
        model = BayesianRidge(max_iter=300)
        model.fit(self.X, self.y)
        return model

# 预测未来值
future_years = np.array([2024, 2025, 2026])
future_X = np.zeros((3, X.shape[1]))

# 预测未来特征值
for i, feature in enumerate(X.columns):
    model = np.poly1d(np.polyfit(data['year'], data[feature], 2))
    future_X[:, i] = model(future_years)

# 初始化模型
ts_models = TimeSeriesModels(y)
```

```
reg_models = RegressionModels(X, y)

# 存储预测结果
predictions = {}
bootstrap_predictions = {}
n_bootstrap = 1000

# 时间序列模型预测
predictions['Holt-Winters'] = ts_models.holt_winters_forecast()
predictions['ExpSmoothing'] = ts_models.exp_smoothing_forecast()
predictions['MovingAverage'] = ts_models.moving_average()

# 回归模型预测
poly_model, poly_transform = reg_models.polynomial_regression()
future_X_poly = poly_transform.transform(future_X)
predictions['Polynomial'] = poly_model.predict(future_X_poly)

lr_model = reg_models.linear_regression()
predictions['Linear'] = lr_model.predict(future_X)

br_model = reg_models.bayesian_ridge()
predictions['BayesianRidge'] = br_model.predict(future_X)

# Bootstrap for confidence intervals
for name in predictions.keys():
    bootstrap_pred = np.zeros((n_bootstrap, len(future_years)))

    for i in range(n_bootstrap):
        indices = np.random.randint(0, len(y), size=len(y))
        y_boot = y.iloc[indices]

        if name in ['Holt-Winters', 'ExpSmoothing', 'MovingAverage']:
            ts_models_boot = TimeSeriesModels(y_boot)
            if name == 'Holt-Winters':
                bootstrap_pred[i, :] = ts_models_boot.holt_winters_forecast()
            elif name == 'ExpSmoothing':
                bootstrap_pred[i, :] = ts_models_boot.exp_smoothing_forecast()
            else:
                bootstrap_pred[i, :] = ts_models_boot.moving_average()
        else:
            X_boot = X.iloc[indices]
            if name == 'Polynomial':
                model_boot, _ = RegressionModels(X_boot,
y_boot).polynomial_regression()
                bootstrap_pred[i, :] = model_boot.predict(future_X_poly)
            elif name == 'Linear':
                model_boot = RegressionModels(X_boot, y_boot).linear_regression()
                bootstrap_pred[i, :] = model_boot.predict(future_X)
            else:
                model_boot = RegressionModels(X_boot, y_boot).bayesian_ridge()
                bootstrap_pred[i, :] = model_boot.predict(future_X)

    bootstrap_predictions[name] = {
        'mean': np.mean(bootstrap_pred, axis=0),
        'lower': np.percentile(bootstrap_pred, 2.5, axis=0),
        'upper': np.percentile(bootstrap_pred, 97.5, axis=0)
```

```

    }

plt.figure(figsize=(20, 15))
for i, (name, pred) in enumerate(predictions.items(), 1):
    plt.subplot(3, 2, i)

    # 绘制历史数据
    plt.plot(data['year'], y, 'o-', label='Historical Data', marker='s', markersize=10,
             markeredgcolor='black', linewidth=3, color='#F2C5C6')

    # 绘制预测值
    plt.plot(future_years, pred, label='Forecast', marker='s', markersize=10,
             markeredgcolor='black', linewidth=3, color='#A8D8B9')

    # 绘制置信区间
    plt.fill_between(future_years,
                    bootstrap_predictions[name]['lower'],
                    bootstrap_predictions[name]['upper'],
                    color='grey',
                    alpha=0.3,
                    label='95% Confidence Interval')

    plt.title(f'{name} Forecast', fontsize=16, fontweight='bold')
    plt.xlabel('Year', fontsize=14)
    plt.ylabel('Export Value (100M USD)', fontsize=14)
    plt.legend(fontsize=12)
    plt.grid(True, linestyle='--', alpha=0.7)
    plt.xticks(fontsize=12)
    plt.yticks(fontsize=12)

plt.tight_layout()
plt.show()
# 修改预测结果的输出部分
print("\n各模型预测结果（单位：亿美元）：")
print("="*80)
print(f'模型名称':<15} {'2024预测':<10} {'2025预测':<10} {'2026预测':<10} {'置
信区间宽度':<15}")
print("="*80)

for name, pred in predictions.items():
    if isinstance(pred, pd.Series):
        pred_values = pred.values
    else:
        pred_values = pred

    interval_width = np.mean(bootstrap_predictions[name]['upper'] -
bootstrap_predictions[name]['lower'])
    print(f'{name}:<15} {pred_values[0]:10.2f} {pred_values[1]:10.2f}
{pred_values[2]:10.2f} {interval_width:15.2f}")

# 集成预测
ensemble_pred = np.mean([
    predictions[name].values if isinstance(predictions[name], pd.Series)
    else predictions[name]
    for name in predictions.keys()

```

```

], axis=0)

ensemble_lower = np.mean([bootstrap_predictions[name]['lower'] for name in
predictions.keys()], axis=0)
ensemble_upper = np.mean([bootstrap_predictions[name]['upper'] for name in
predictions.keys()], axis=0)

print("\n集成模型预测结果: ")
print("="*50)
for year, pred, lower, upper in zip(future_years, ensemble_pred, ensemble_lower,
ensemble_upper):
    print(f'{year} 年: {pred:.2f} [{lower:.2f}, {upper:.2f}]')

# 计算预测增长率
print("\n各模型预测增长率: ")
print("="*80)
for name, pred in predictions.items():
    print(f'\n{name}:')
    if isinstance(pred, pd.Series):
        pred_values = pred.values
    else:
        pred_values = pred
    growth_rates = np.diff(pred_values) / pred_values[:-1] * 100
    for year, rate in zip(future_years[1:], growth_rates):
        print(f'{year} 年: {rate:.1f}%')

# 计算模型在历史数据上的表现
print("\n模型在历史数据上的表现 (RMSE): ")
print("="*50)
for name, pred in predictions.items():
    if name in ['Holt-Winters', 'ExpSmoothing', 'MovingAverage']:
        ts_models = TimeSeriesModels(y)
        if name == 'Holt-Winters':
            historical_pred = ts_models.holt_winters_forecast()
        elif name == 'ExpSmoothing':
            historical_pred = ts_models.exp_smoothing_forecast()
        else:
            historical_pred = ts_models.moving_average()
    else:
        if name == 'Polynomial':
            historical_pred = poly_model.predict(poly_transform.transform(X))
        elif name == 'Linear':
            historical_pred = lr_model.predict(X)
        else:
            historical_pred = br_model.predict(X)

    if isinstance(historical_pred, pd.Series):
        historical_pred = historical_pred.values

    rmse = np.sqrt(np.mean((y.values - historical_pred[:len(y)]) ** 2))
    print(f'{name:<15} {rmse:10.2f}')

```

```
import pandas as pd
import numpy as np

# 1. 创建市场预测数据框架
def create_forecast_df():
    # 中国宠物数量预测
    pet_forecast = {
        'Year': [2024, 2025, 2026],
        'Cat_Population': [7760.8, 8441.5, 9122.2],
        'Dog_Population': [5081.5, 5021.6, 4961.6],
        'Total_Pets': [12842.3, 13463.1, 14083.8] # 计算总量
    }

    # 各国宠物食品需求预测（单位：亿美元）
    demand_forecast = {
        'Year': [2024, 2025, 2026],
        'China_Demand': [21.52, 22.83, 24.13],
        'US_Demand': [43.55, 44.89, 46.22],
        'France_Demand': [3.47, 3.63, 3.79],
        'Germany_Demand': [3.97, 4.15, 4.34]
    }

    # 中国生产和出口预测（单位：万吨）
    production_export = {
        'Year': [2024, 2025, 2026],
        'Production': [2762.99, 3792.44, 3861.00],
        'Export': [32.27, 44.63, 42.29],
        'Domestic_Supply': [2730.72, 3747.81, 3818.71] # 计算国内供应量
    }

    # 关税政策数据
    tariff_data = {
        'Country': ['US', 'EU', 'Japan', 'Korea'],
        'Tariff_Rate': [0.04, 0.04, 0.04, 0.04],
        'Previous_Rate': [0.15, 0.04, 0.04, 0.04], # 美国之前是15%
        'Quota_Restriction': ['None', 'None', 'None', 'None']
    }

    # 创建DataFrame
    df_pets = pd.DataFrame(pet_forecast)
    df_demand = pd.DataFrame(demand_forecast)
    df_production = pd.DataFrame(production_export)
    df_tariff = pd.DataFrame(tariff_data)

    return {
        'pets': df_pets,
        'demand': df_demand,
        'production': df_production,
        'tariff': df_tariff
    }

# 创建数据框架
dfs = create_forecast_df()

# 打印数据框架预览
```

```
print("\n宠物数量预测:")
print(dfs['pets'])
print("\n需求预测:")
print(dfs['demand'])
print("\n生产和出口预测:")
print(dfs['production'])
print("\n关税政策:")
print(dfs['tariff'])

import pandas as pd
import numpy as np
from scipy import stats

class TariffImpactAnalysis:
    def __init__(self, dfs):
        self.dfs = dfs

    def calculate_tariff_elasticity(self):
        """
        计算关税弹性
        关税弹性 = (出口量变化率)/(关税税率变化率)
        """
        # 美国关税从15%降到4%的影响分析
        us_tariff_change = (0.04 - 0.15) / 0.15 # -73.33%

        # 计算2024-2026年的出口增长率
        export_2024 = self.dfs['production']['Export'][0]
        export_2026 = self.dfs['production']['Export'][2]
        export_growth = (export_2026 - export_2024) / export_2024

        # 计算关税弹性
        tariff_elasticity = export_growth / us_tariff_change

        return {
            'tariff_change_pct': us_tariff_change * 100,
            'export_growth_pct': export_growth * 100,
            'elasticity': tariff_elasticity
        }

    def analyze_market_impact(self):
        """
        分析各主要市场的影响
        """
        # 计算各市场需求增长率
        market_growth = {}
        for column in self.dfs['demand'].columns:
            if column != 'Year':
                growth = (self.dfs['demand'][column].iloc[-1] -
                        self.dfs['demand'][column].iloc[0]) /
self.dfs['demand'][column].iloc[0]
                market_growth[column] = growth * 100

        return market_growth

    def calculate_policy_sensitivity(self):
```

```

"""
    计算政策敏感度指标
    """
    # 计算出口占总产量比例
    export_ratio = self.dfs['production']['Export'] / self.dfs['production']['Production']

    # 计算各年度政策敏感度
    sensitivity = {
        'export_ratio': export_ratio.mean() * 100,
        'export_volatility': export_ratio.std() * 100,
        'market_concentration': self.calculate_market_concentration()
    }

    return sensitivity

def calculate_market_concentration(self):
    """
    计算市场集中度（基于主要出口市场需求份额）
    使用赫芬达尔指数(HHI)
    """
    total_demand = self.dfs['demand'].iloc[0][['US_Demand', 'France_Demand',
        'Germany_Demand']].sum()
    market_shares = self.dfs['demand'].iloc[0][['US_Demand', 'France_Demand',
        'Germany_Demand']] / total_demand
    hhi = (market_shares ** 2).sum() * 10000

    return hhi

# 创建分析实例
analysis = TariffImpactAnalysis(dfs)

# 获取分析结果
elasticity_results = analysis.calculate_tariff_elasticity()
market_impact = analysis.analyze_market_impact()
policy_sensitivity = analysis.calculate_policy_sensitivity()

# 打印分析结果
print("\n关税弹性分析结果:")
print(f'关税变化率: {elasticity_results['tariff_change_pct']:.2f}%')
print(f'出口增长率: {elasticity_results['export_growth_pct']:.2f}%')
print(f'关税弹性系数: {elasticity_results['elasticity']:.2f}')

print("\n市场增长率分析:")
for market, growth in market_impact.items():
    print(f'{market}: {growth:.2f}%')

print("\n政策敏感度指标:")
print(f'平均出口比例: {policy_sensitivity['export_ratio']:.2f}%')
print(f'出口波动性: {policy_sensitivity['export_volatility']:.2f}%')
print(f'市场集中度(HHI): {policy_sensitivity['market_concentration']:.0f}')

# 进行影响程度评估
def assess_impact_level(elasticity, market_concentration, export_ratio):
    """评估政策影响程度"""

```



```
    impact_score = 0
    impact_factors = []

    # 评估关税弹性影响
    if abs(elasticity) > 1:
        impact_score += 3
        impact_factors.append("高关税敏感度")
    elif abs(elasticity) > 0.5:
        impact_score += 2
        impact_factors.append("中等关税敏感度")
    else:
        impact_score += 1
        impact_factors.append("低关税敏感度")

    # 评估市场集中度影响
    if market_concentration > 2500:
        impact_score += 3
        impact_factors.append("高市场集中度")
    elif market_concentration > 1500:
        impact_score += 2
        impact_factors.append("中等市场集中度")
    else:
        impact_score += 1
        impact_factors.append("低市场集中度")

    # 评估出口依赖度
    if export_ratio > 20:
        impact_score += 3
        impact_factors.append("高出口依赖")
    elif export_ratio > 10:
        impact_score += 2
        impact_factors.append("中等出口依赖")
    else:
        impact_score += 1
        impact_factors.append("低出口依赖")

    # 综合评估
    total_impact = "高" if impact_score >= 7 else "中" if impact_score >= 5 else "低"

    return {
        'impact_level': total_impact,
        'impact_score': impact_score,
        'key_factors': impact_factors
    }

# 进行影响程度评估
impact_assessment = assess_impact_level(
    elasticity_results['elasticity'],
    policy_sensitivity['market_concentration'],
    policy_sensitivity['export_ratio']
)

print("\n政策影响程度评估:")
print(f"影响程度: {impact_assessment['impact_level']}")
```

```

print(f'影响得分: {impact_assessment['impact_score']/9}')
print("关键影响因素:")
for factor in impact_assessment['key_factors']:
    print(f'- {factor}')

class DetailedPolicyImpactAnalysis:
    def __init__(self, dfs):
        self.dfs = dfs

    def analyze_direct_impacts(self):
        """分析直接影响"""
        # 计算关税成本影响
        base_export = self.dfs['production']['Export'][0] # 2024年出口量
        us_market_share = 0.40 # 假设美国市场占出口份额40%

        old_tariff_cost = base_export * us_market_share * 0.15 # 旧关税成本
        new_tariff_cost = base_export * us_market_share * 0.04 # 新关税成本
        tariff_cost_saving = old_tariff_cost - new_tariff_cost

        return {
            'tariff_cost_reduction': tariff_cost_saving,
            'cost_saving_ratio': (tariff_cost_saving / old_tariff_cost) * 100
        }

    def analyze_market_competitiveness(self):
        """分析市场竞争力变化"""
        # 计算主要市场的价格竞争力指数
        markets = ['US_Demand', 'France_Demand', 'Germany_Demand']
        competitiveness = {}

        for market in markets:
            growth_rate = (self.dfs['demand'][market].iloc[-1] -
                           self.dfs['demand'][market].iloc[0]) / self.dfs['demand'][market].iloc[0]
            market_share = self.dfs['demand'][market].iloc[0] /
self.dfs['demand'].iloc[0][markets].sum()

            competitiveness[market] = {
                'growth_rate': growth_rate * 100,
                'market_share': market_share * 100,
                'competitive_index': growth_rate * market_share * 100
            }

        return competitiveness

    def analyze_industry_structure_impact(self):
        """分析产业结构影响"""
        # 计算产业结构变化指标
        production = self.dfs['production']

        domestic_ratio = (production['Production'] - production['Export']) /
production['Production']
        export_ratio = production['Export'] / production['Production']

        return {
            'domestic_ratio': domestic_ratio.mean() * 100,

```

```

        'export_ratio': export_ratio.mean() * 100,
        'structure_change': {
            'year_2024': export_ratio.iloc[0] * 100,
            'year_2026': export_ratio.iloc[-1] * 100,
            'change': (export_ratio.iloc[-1] - export_ratio.iloc[0]) * 100
        }
    }

def calculate_risk_exposure(self):
    """计算风险敞口"""
    # 计算不同维度的风险敞口
    production = self.dfs['production']

    # 计算产能利用率波动
    capacity_utilization = production['Export'] / production['Production']
    capacity_volatility = capacity_utilization.std() * 100

    # 计算市场集中度风险
    market_concentration = self.dfs['demand'].iloc[0][['US_Demand',
'France_Demand', 'Germany_Demand']]
    hhi = ((market_concentration / market_concentration.sum()) ** 2).sum() * 10000

    return {
        'capacity_risk': capacity_volatility,
        'market_concentration_risk': hhi,
        'overall_risk_level': self._assess_risk_level(capacity_volatility, hhi)
    }

def _assess_risk_level(self, capacity_risk, concentration_risk):
    """评估综合风险水平"""
    risk_score = 0

    # 评估产能风险
    if capacity_risk > 15:
        risk_score += 3
    elif capacity_risk > 10:
        risk_score += 2
    else:
        risk_score += 1

    # 评估集中度风险
    if concentration_risk > 2500:
        risk_score += 3
    elif concentration_risk > 1500:
        risk_score += 2
    else:
        risk_score += 1

    return "高" if risk_score >= 5 else "中" if risk_score >= 3 else "低"

# 创建分析实例并执行分析
detailed_analysis = DetailedPolicyImpactAnalysis(dfs)

# 获取各项分析结果
direct_impacts = detailed_analysis.analyze_direct_impacts()

```

```

market_competitiveness = detailed_analysis.analyze_market_competitiveness()
industry_structure = detailed_analysis.analyze_industry_structure_impact()
risk_exposure = detailed_analysis.calculate_risk_exposure()

# 打印详细分析结果
print("\n=== 政策影响深度分析 ===")

print("\n1. 直接成本影响:")
print(f'关税成本节省比例: {direct_impacts['cost_saving_ratio']:.2f}%')

print("\n2. 市场竞争力分析:")
for market, metrics in market_competitiveness.items():
    print(f'\n{market}:')
    print(f'增长率: {metrics['growth_rate']:.2f}%')
    print(f'市场份额: {metrics['market_share']:.2f}%')
    print(f'竞争力指数: {metrics['competitive_index']:.2f}')

print("\n3. 产业结构影响:")
print(f'平均内销比例: {industry_structure['domestic_ratio']:.2f}%')
print(f'平均出口比例: {industry_structure['export_ratio']:.2f}%')
print(f'出口比例变化: {industry_structure['structure_change']['change']:.2f}%')

print("\n4. 风险敞口评估:")
print(f'产能利用波动风险: {risk_exposure['capacity_risk']:.2f}%')
print(f'市场集中度风险指数: {risk_exposure['market_concentration_risk']:.0f}')
print(f'综合风险水平: {risk_exposure['overall_risk_level']}')

# 生成综合影响评估报告
def generate_impact_summary():
    impact_levels = {
        'cost_impact': '高' if direct_impacts['cost_saving_ratio'] > 50 else '中' if
direct_impacts['cost_saving_ratio'] > 25 else '低',
        'market_impact': '高' if any(m['competitive_index'] > 50 for m in
market_competitiveness.values()) else '中' if any(m['competitive_index'] > 25 for m
in market_competitiveness.values()) else '低',
        'structure_impact': '高' if abs(industry_structure['structure_change']['change']) >
10 else '中' if abs(industry_structure['structure_change']['change']) > 5 else '低',
        'risk_level': risk_exposure['overall_risk_level']
    }

    return impact_levels

impact_summary = generate_impact_summary()

print("\n=== 综合影响评估 ===")
print(f'成本影响程度: {impact_summary['cost_impact']}')
print(f'市场影响程度: {impact_summary['market_impact']}')
print(f'结构影响程度: {impact_summary['structure_impact']}')
print(f'风险影响程度: {impact_summary['risk_level']}')

```