Team Number :	apmcm24207642
Problem Chosen :	C

2024 APMCM summary sheet

Pet Food Industry Development Strategies Based on Time Series Tree Models

Abstract

The pet industry is thriving globally, with pet expenses becoming a significant part of consumption. This article explores the past five years' development of the pet industry in China and worldwide, forecasts trends for the next three years, and proposes strategies for sustainable pet food industry growth in China.

For Question 1, analyze the historical data through **data visualization**. In addition, **Spearman coefficient** is used to analyze whether there was a significant correlation between various factors and the development of China's pet industry. After that, **ARIMA model** is used to predict the data of the factors in the next three years. Eventually, **Random Forest Model** and **Decision Tree Regression Model** are compared and optimized to predict the data of China's pet market.

For Question 2, considering the global scope but focusing on key factors similar to Question 1, due to data collection challenges, we **use data from China, the US, Germany, France, and Japan as representatives.** Predicting the global demand for pet food in the next three years, which **is equivalent to the sales volume of pet food** in the next three years, making data collection and calculation easier and more efficient.

For Question 3, our team collect relevant data such as the proportion of China's exports in the world market, and obtain the development situation of China's pet food industry. Following the ideas and methods of Question 1 and Question 2 , the production and export of pet food in China in the next three years are obtained.

For Question 4, **a model of price elasticity of demand** is established to better reflect the change of demand with the change of price, so as to reflect the change of production and export value. (This paper **considers the degree of change in the production and export value of pet food in China as a benchmark** to develop the strategy). Moreover, the sensitivity analysis of tariff is carried out, the range of its change is limited to less than 10%, the change of production and export value is observed, and the image meaning analysis is combined to give the strategy.

Key Words ARIMA model Random forest regression model Decision tree regression model

Content

I. Introduction

1.1 Problem Background

Due to the rapid development of economy and the improvement of per capita income, people's consumption concept is constantly developing, the pet industry is gradually emerging, and pet-related expenses are increasingly becoming a non-negligible part of people's consumption. With the rise of the concept of "pet companionship" in China, pet-related industries such as pet food[1], pet clinics, pet supplies and pet care have also formed a large and fast-growing market.

1.2 Restatement of the Problem

Based on the data provided in the title attachment and additional data sought by the team, this paper will conduct data processing and model establishment and solution to solve the following four problems.

- **Question 1**: Analyze the development of China's pet industry in the past five years according to pet types; And analyze the factors that affect the development of China's pet industry, and then make a suitable mathematical model according to these factors to predict the development of China's pet industry in the next three years.
- **Question 2**: Expand from the situation in China to the world, analyze the development of the global pet industry, and make a suitable mathematical model to predict the global demand for pet food in the next three years.
- **Question 3**: Based on the results of the previous two questions, analyze the development of China's pet food industry and forecast the production and export of China's pet food in the next three years (regardless of economic policy changes).
- **Question 4**: Based on the above analysis, an appropriate mathematical model is constructed to formulate feasible strategies for the sustainable development of China's pet food industry.

II. Problem analysis

2.1 Analysis of Question 1

Question 1 can be divided into three parts. The first part is to analyze China's pet industry over the past five years by pet type, focusing on cats and dogs due to their popularity. Collect data on average annual consumption for these pets and consider pet medical care, supplies, and food industries. The second part is to analyze factors influencing China's pet industry, considering both external (economic development, per capita GDP, urbanization) and internal (aging population, singles) factors.The last one is to predict China's pet industry development in the next three years by forecasting factor data and using the relationship between these factors and pet industry development obtained in the second part.

2.2 Analysis of Question 2

Question 2 can be split into two parts.The first part extends the analytical thinking of the first question globally.The second part aims to predict global pet food demand over the next three years by first forecasting the global pet industry trend and then searching specific pet food data (dry, wet, and other categories) for detailed predictions.

2.3 Analysis of Question 3

Question 3 can be divided into two parts.Firstly,analyze China's pet food industry based on global pet food demand trends and China's development situation. Focus on domestic market demand, production and supply capacity, and export market performance. Domestic demand includes analyzing pet food demand and trends. Production and supply consider domestic pet food production volumes, enterprise numbers, and product structures. Export performance assesses China's competitiveness in the global pet food market.Secondly,forecast China's pet food production and exports, summing these to get the final forecast. Domestic demand, driven by pet numbers, per capita consumption, and market size, and global demand, considering China's export share, drive production and exports.

2.4 Analysis of Question 4

For Question 4, an appropriate mathematical model is constructed to formulate feasible strategies for the sustainable development of China's pet food industry. In order to complete this question better, we combine the results of the above three questions, consider the impact of tariff policy, analyze the changes in pet food output and production under its impact, and then get the final strategy.

The whole idea of problem C is shown in Figure 1：

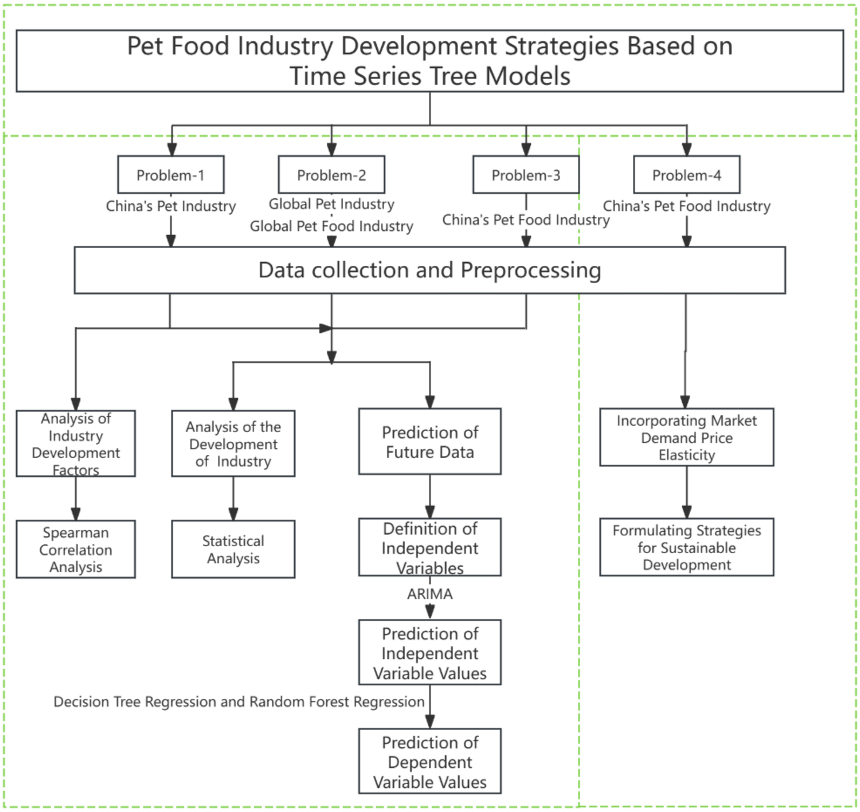


Figure 1

III. Model hypothesis

1. People's preference for pets will not change significantly in recent years.

2. In recent years, no other products have posed a major threat to the pet food market.

3. There will be no economic crisis in the next few years.

4. In the next few years, there will be no disasters such as epidemics that will have a huge impact on the population.

5. The data obtained are accurate and valid.

IV. Symbol description

Symbol	Implication	Unit
β_0	Intercept term	/
β_i	i th regression coefficient，i=1,2,…	/
X_6	Random error term	/
y_t	The value of the time series at the Current time t	/
$y_{t-1}, y_{t-2} \cdots$	Lagged value	/
$\varphi_1, \varphi_2, \ldots$	Random error at time t	/
ϵ_t	AR	/
$\epsilon_{t-1}, \epsilon_{t-2} \cdots$	Residual value of historical Predictions	/
$\theta_1,\ \theta_2 \ldots$	MA	/
μ	Constant term	/
e_d	Elastic coefficient of China	/
e_d'	Elastic coefficient of America	/
α	Tariff rate	/

V. the Establishment and Solution of the Model

5.1 Establishment and solution of Problem-1 model

5.1.1 Data collection and Preprocessing

Since this topic needs to analyze and forecast the development of China's pet industry, but the attached data is limited, so data collection should be carried out first. Consider data collection from the number of cats and dogs, GDP per capita, pet market size and other dimensions.

After data collection, data preprocessing is carried out.

Step1 Missing value and outlier test This paper uses Excel to screen the data, and no missing value is found. Then draw a box diagram in Python to find whether there are outliers. As a result, it is found that most of the dimensions are within the normal range, and only a few of the dimensions have a few outliers. Since this paper is conducted under a small sample, and the anomalies of some data may involve some regularities, this paper will not delete the outliers.

Step2 Data standardization As there is a big difference in the dimensions of existing data and collected data, in order to facilitate subsequent analysis and model invocation, data standardization is carried out here. The formula is as follows:

(Because the amount of data is so large, the standardized data is saved as files in supporting materials.)

5.1.2 Statistical analysis of the development of China's pet industry

Since there are many dimensions of data collected in this paper, data profiling is considered first to visually demonstrate the development situation, as shown in the Figure 2、Figure 3、Figure 4:

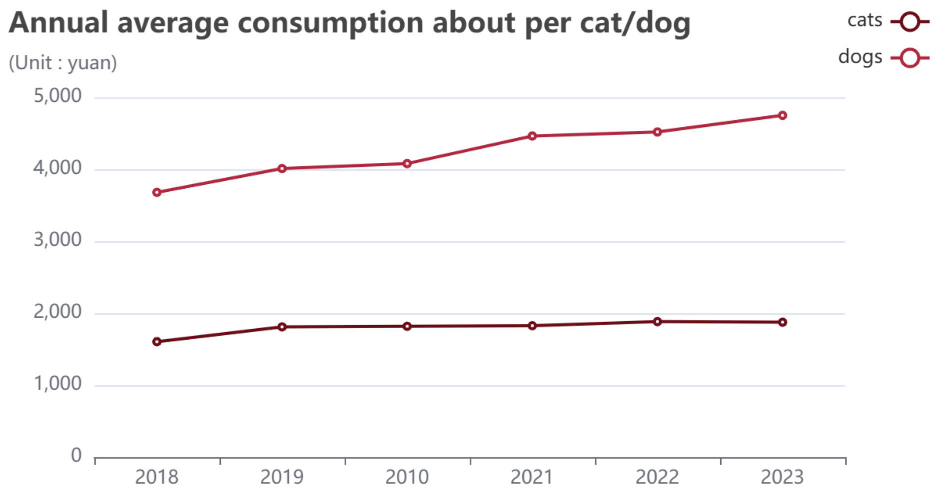


Figure 2

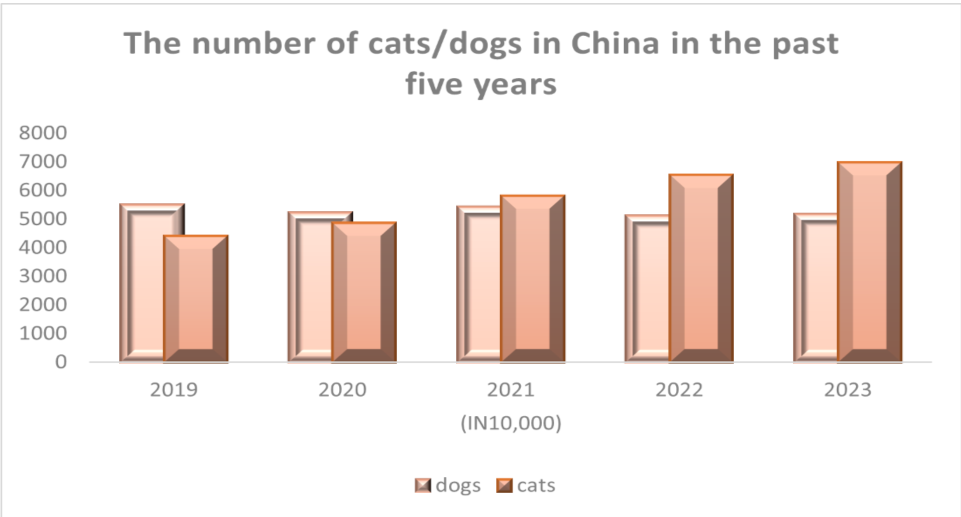


Figure 3

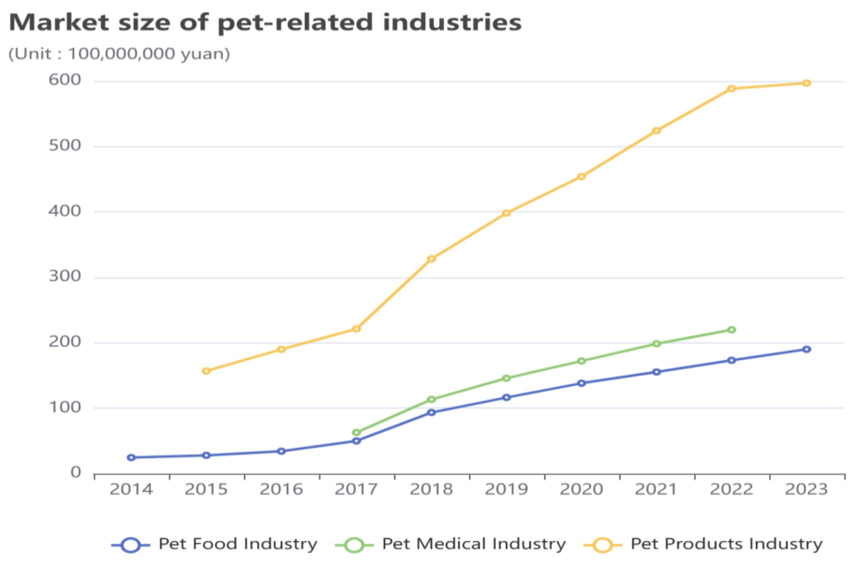


Figure 4

Through the above data, we can preliminarily see that the development of the pet industry in China shows a steady upward trend. For example, from 2019 to 2020, the size of the pet food market will increase by 18.7% year-on-year, and the average annual expenditure of a single dog will increase by 9.07% year-on-year from 2022 to 2023.

In addition, we can also calculate the compound annual growth rate of cats and dogs in China in the past five years, using the following formula:

$$CAGR = \left(\frac{N_t}{N_0}\right)^{\frac{1}{t}} - 1 \tag{1}$$

Where, N_t is the number of cats(dogs) in 2023, and N_0 is the number of cats/dogs in 2019. The calculation results are shown in Table 1：

Table 1

Cats	0.10
Dogs	-0.01

Despite a slight decline in the number of dogs, the overall number of cats and dogs in China has continued to show an upward trend in the past five years due to an even greater increase in the number of cats.

Descriptive statistics were performed on the data of each dimension, as shown in the

Table 2:

Table 2

variable	average value	standard deviation	median	variance	Kurtosis	Variation coefficient
The number of cats	5289.6	167.17	5222.0	-2.41	0.503	0.032
The number of dogs	5719.2	1084.89	5806.0	-2.20	-0.104	0.190

(See Appendix for full data analysis results)

As can be seen from the above table, in the past five years, the number of cats and dogs has been more than 50 million, of which the number of cats is relatively stable, and the number of dogs has been significantly fluctuating. The number of Kurtosis of cats and dogs is less than zero, indicating that their numbers are relatively evenly distributed.

Conclusion From the above analysis results, it can be seen that the development of China's pet industry in the past five years shows an upward trend according to the two pet types of cats and dogs. The market size of pet food, supplies, medical [3]treatment, etc. continues to increase, and the total number of pets continues to rise.

5.1.3 Establishment of a model based on multiple linear regression to explore the development factors of China's pet industry

In order to simplify the research model, this paper first quantifies the development of China's pet industry, taking the total number of pets as a measure, because the development of the pet industry is closely related to the overall number of pets.

In order to study which factors can affect the development of the pet industry, data analysis is considered from the dimensions of per capita GDP, urbanization rate, aging ratio, etc., to explore whether they have an impact on the development of the pet industry.

For the collected data, this topic still uses data visualization to their trends, as shown in the Figure 5:

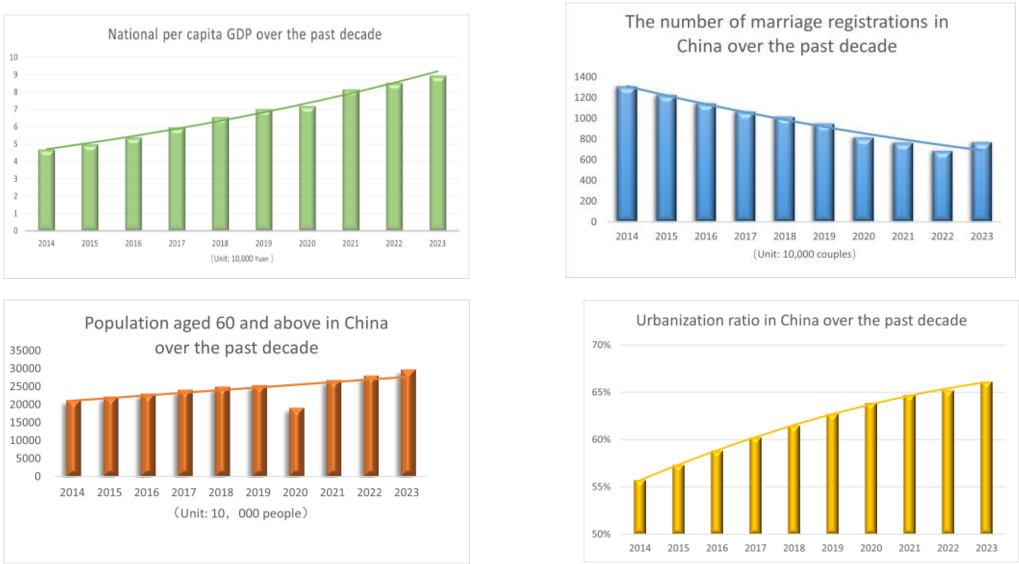


Figure 5

As can be seen from the above four figures, the per capita GDP and urbanization rate of China have both increased steadily in the past five years, indicating that Chinese people have more disposable property, which will increase the probability of Chinese people spending on pets. In addition, the increasing proportion of aging and the declining number of marriages mean that more people need companionship, which is also in line with the concept of "pet companionship"[2][2], and the number of pet owners will continue to increase.

5.1.4 A model solution for exploring the development factors of China's pet industry based on multiple linear regression

Spearman correlation coefficient is considered to be used for judgment, and the specific results are shown in the Figure 6 :

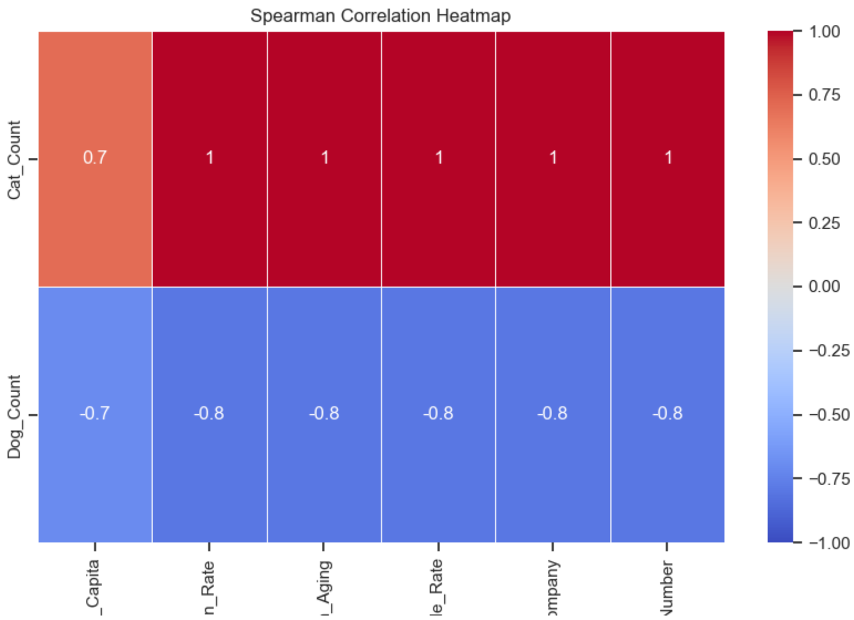


Figure 6

Based on Figure 6, it can be observed that the development of the pet industry has a close correlation with the indicators we searched for. Except for per capita GDP, the correlation coefficients between the other five indicators and the number of cats are all 1, while the correlation coefficients between these indicators and the number of dogs are all 0.8.

5.1.5 Based on the ARIMA model, a prediction model for the total number of pets in random forest regression is established

Based on the conclusions of the multiple linear regression model, we adopt... As the independent variable, the total number of pets was taken as the dependent variable for further study.

First, the predicted values of the independent variables for the next three years are calculated using a time series model, considering their continuity and random variability over time. The ARIMA model is adopted for this purpose. The specific steps involve utilizing the model to forecast the independent variable data for the upcoming three years.

Step1 Stationarity test The time series[4] diagram (Figure 7) is made to observe whether there are outliers. If there are outliers, the data needs to be transformed, or the difference method is used to process the data.

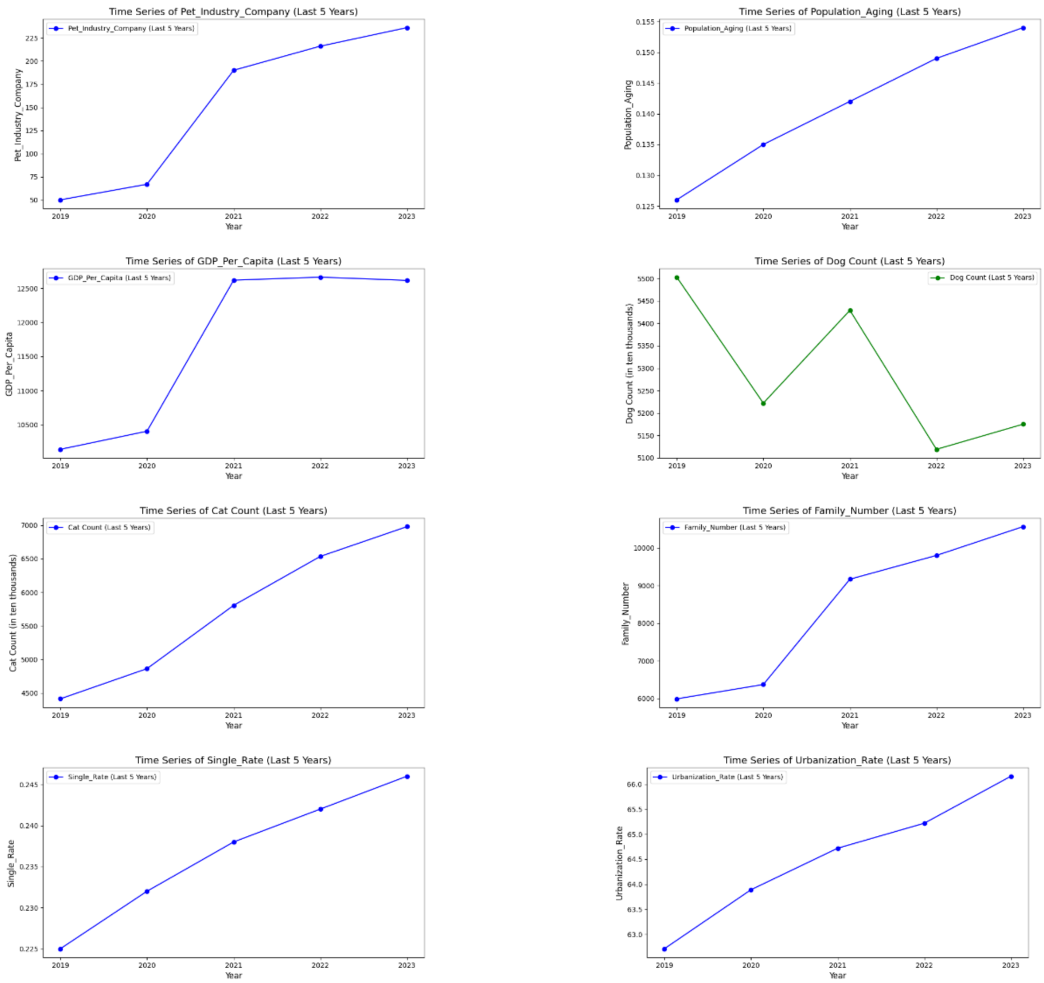
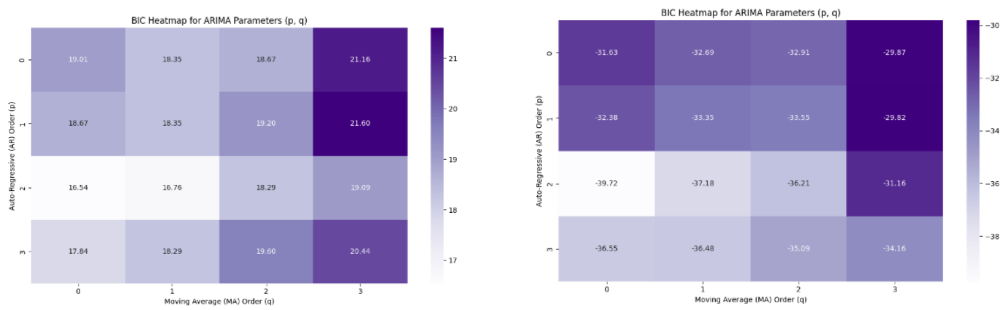


Figure 7

From the figure mentioned above, it can be observed that the time series [5][5]of the independent variable is stationary, allowing for direct progression to the next step of the operation.

Step2 Determine the autoregressive correlation coefficient p and partial regression correlation coefficient q AIC and BIC grid search methods are used to determine the most suitable p and q values. p-q calculation results of six factors are shown in Figure 8:



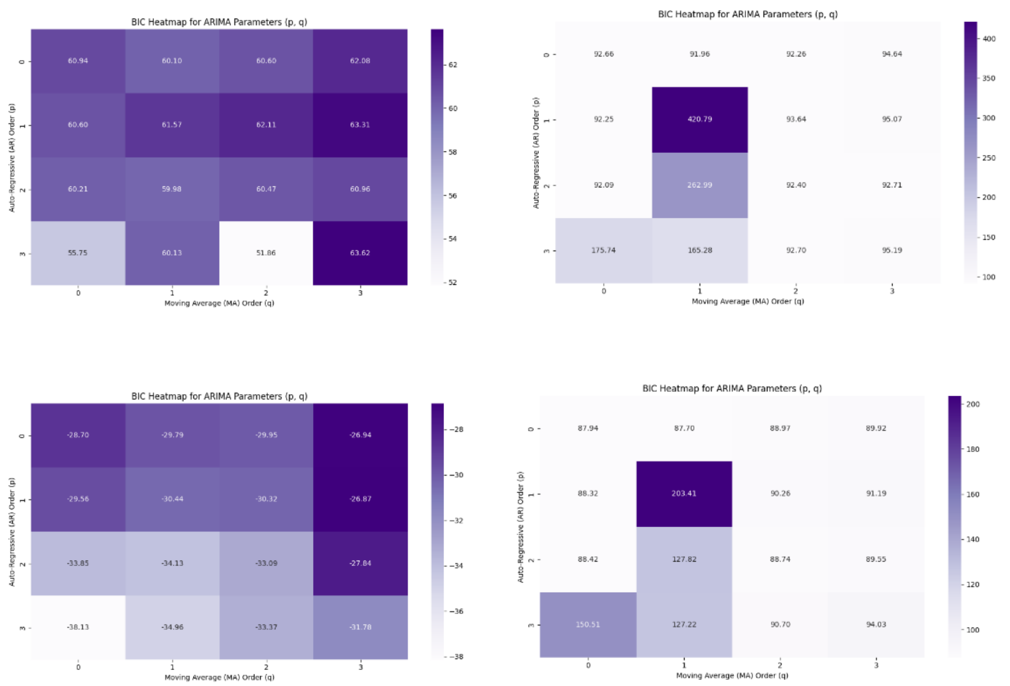


Figure 8

The best p values and q values of each variable are shown in the Table 3

Table 3

variable	Per capita GDP	Proportion of the aging population	Urbanization rate	Proportion of singles	Number of pet companies	Number of households owning pets
p	0	3	2	2	3	2
d	1	0	0	0	2	0

Step3 Carry out residual white noise test. (Q test)

Table 4 Result of the Q-test

Model Statistics			
Variable	Statistics	p-value	DF
Per capita GDP	2.758305248	0.599051181	4
Proportion of the aging population	2.058348943	0.725027695	4
Urbanization rate	2.793761962	0.592909989	4
Proportion of singles	2.012204223	0.733514055	4
Number of pet companies	2.427527121	0.65765847	4
Number of households owning pets	2.06307426	0.724158914	4

The p-values from the Q-test on the residuals of six variables exceed 0.05, failing to reject the null hypothesis that they are white noise sequences. Thus, the residuals of Per capita GDP, aging population proportion, urbanization rate, singles proportion, pet companies count, and pet-owning households are confirmed as white noise sequences.

Step4 Establish ARIMA (p,d,q) model

Table 5 Results ofARIMA model

Model Statistics		
	Variable	Model Type
Model ID	Per capita GDP	ARIMA(0,0,1)
	Proportion of the aging population	ARIMA(3,0,0)
	Urbanization rate	ARIMA(2,0,0)
	Proportion of singles	ARIMA(2,0,0)
	Number of pet companies	ARIMA(3,0,2)
	Number of households owning pets	ARIMA(2,0,0)

The calculation formula is as follows.:

Per capita GDP

$$y_t = \mu + \theta_1 \epsilon_{t-1} + \epsilon_t \tag{2}$$

Proportion of the aging population

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \varphi_3 y_{t-3} + \epsilon_t \tag{3}$$

Urbanization rate

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \epsilon_t \tag{4}$$

Proportion of singles

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \epsilon_t \tag{5}$$

Number of pet companies

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \varphi_3 y_{t-3} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \epsilon_t \tag{6}$$

Number of households owning pets

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \epsilon_t \tag{7}$$

y_t represents the value of the time series at the current time t,y_{t-1} and y_{t-2} represent Lagged value, φ_1 、 φ_2 … represent Random error at time t,ϵ_t represents AR,

ϵ_{t-1} 、 ϵ_{t-2}… represent Residual value of historical predictions, θ_1 and θ_2

represent MA.

After establishing the ARIMA model to predict the independent variables for the next three years, this paper employs decision tree regression and random forest regression to forecast the dependent variables. Given the small sample size, decision tree regression is chosen due to its suitability, while random forest regression is also considered. However, to mitigate overfitting and noise, a combined approach of comparing and optimizing these two models is adopted for prediction.

The essence of decision tree regression model is to divide the independent variable space into several disjoint regions by recursively splitting the data set, and to represent the target variable with a constant value in each region.

The formula is as follows:

$$y_t' = \sum_{i=1}^{N} c_i * I(x \in R_i) \tag{8}$$

Where y_t' is the predicted value of the target variable (such as the number of pets) in time, N is the total number of regions divided by the decision tree, c_i is the predicted value of the region, x is the per capita GDP, urbanization rate and other feature vectors, and R_i is the first region. Random forest regression model is an integrated model based on decision tree regression. It can improve the stability and generalization ability of the model by constructing multiple decision trees and averaging the prediction results of these trees. The formula is as follows:

$$y'' = \frac{1}{M}\sum_{j=1}^{M} T_j(x) \tag{9}$$

Where y'' is the predicted value of the target variable (such as the number of pets) in time t, and M represents the total number of decision trees[7] in the random forest. $T_j(x)$ represents the predicted value of the i decision tree for the input sample, and x is the feature vector such as per capita GDP and urbanization rate.

5.1.6 Solving the prediction model of total pet population of random forest[6] regression based on ARIMA model

Using Python to program solutions, the predicted value of the independent variable in the next three years is obtained as shown in the

Table 6:

Table 6

Year	Per capita GDP(unit: dollar)	Proportion of the aging population	Urbanization rate	Proportion of singles	Number of pet companies(unit: 10,000s)	Number of households owning pets
2024	13847.6	16.2%	67.01%	25.2%	308	12153
2025	14567.2	16.9%	67.83%	25.7%	360	13411
2026	15286.8	17.6%	68.65%	26.3%	412	14670

Since this paper uses decision tree regression and random forest regression for prediction, the setting of hyperparameters is particularly important when solving Python programming. In this paper, by setting the set of hyperparameters to be valued, the grid search optimal method is adopted to find the hyperparameters, and the hyperparameters are obtained as shown in Table 7:

Table 7

	DTR（cat）	DTR(dog)	RFR(cat)	RFR(dog)
Max depth	4	4	4	4
Min samples split	2	3	2	2
Min samples leaf	1	1	1	1
N estimators	/	/	50	50
Max features	/	/	sqrt	sqrt
Bootstrap	/	/	negative	negative

Finally, Using Python to program solutions, we found that the number of dogs predicted by decision tree regression is good, and the number of cats predicted by random forest regression is good, and the prediction results are as follows: In 2024, 2025, and 2026, the number of cats will be 71,886,900, 75,041,700, and 78,196,300, while the number of dogs will be 51,135,200, 51,708,300, and 51,188,200. The forecast trend chart is shown in Figure 9

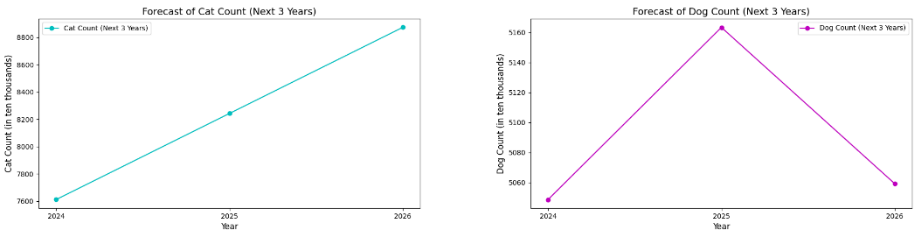


Figure 9

Verify Through calculation, it is found that the determination coefficient is 0.98 and 0.79 respectively, indicating that the fitting effect is good.

5.2 Establishment and solution of Problem-2 model

5.2.1 Data collection and Preprocessing

This topic analyzes global pet industry development and pet food demand, requiring data collection due to limited information. Extending research from China to the world involves similar data considerations as the first question. Additionally, data on cat and dog food sales volumes and annual growth rates of pet populations are necessary. To streamline data search and simplify models, consider selecting representative countries such as China, the United States, France, Germany, and Japan for global analysis.

After collecting the data, it is necessary to carry out data preprocessing work, the general process is similar to the first question, here will not be repeated, we will store the standardized data as a file in the support material.

5.2.2 Data Visualization

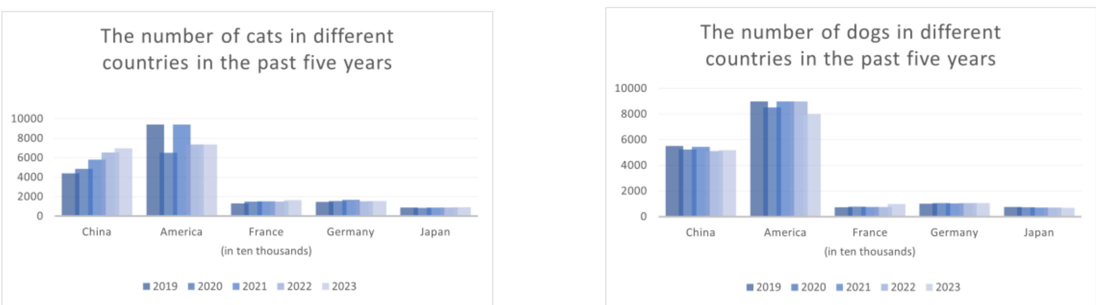


Figure 10

As can be seen from Figure 10and , the total global population of cats and dogs has been relatively stable in the past five years, with the overall population of cats showing an upward trend and the population of dogs declining slightly.

Figure 11 and Figure 12show the distribution of cats and dogs by region:

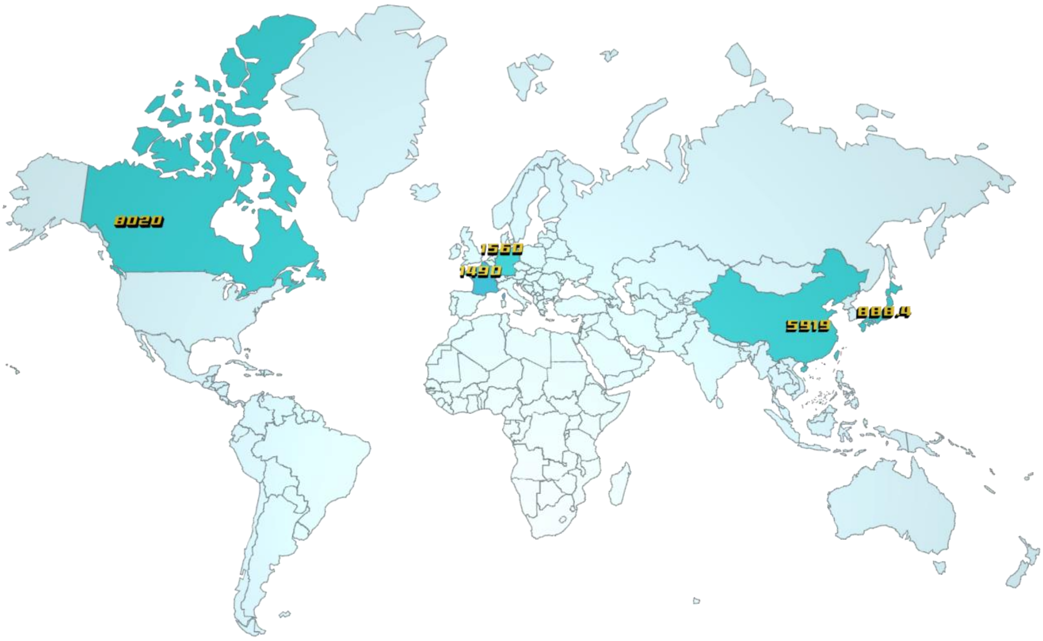


Figure 11

The average number of pet cats in five different countries over the past five years

(unit: ten thousand)

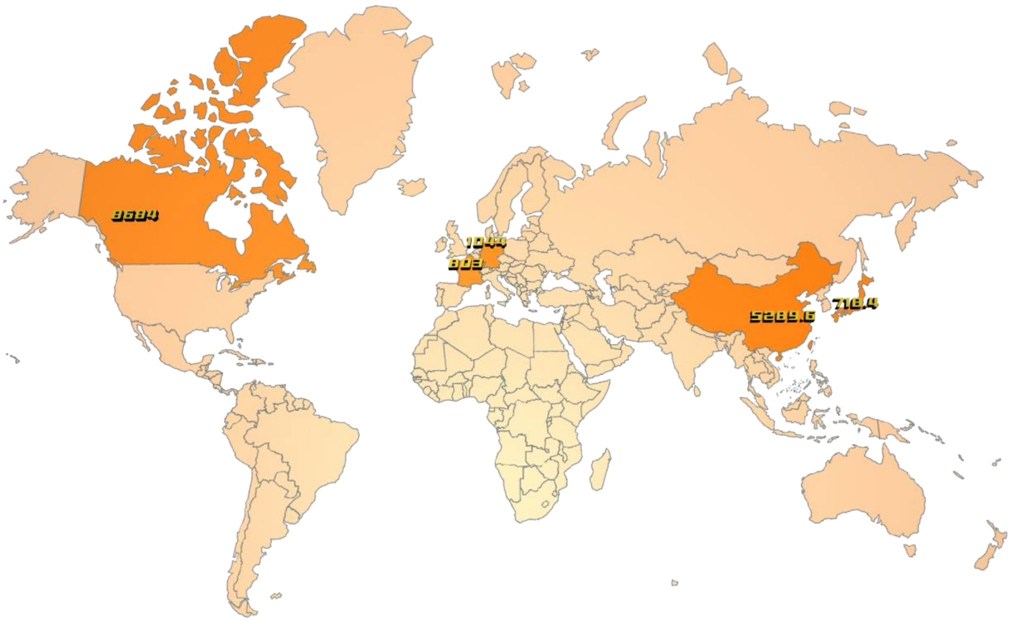


Figure 12

The average number of pet dogs in five different countries over the past five years

(unit: ten thousand)

It can be seen that the number of cats and dogs in China, the United States and other countries is significantly higher than that in other countries, which is related to their large land area and other factors.

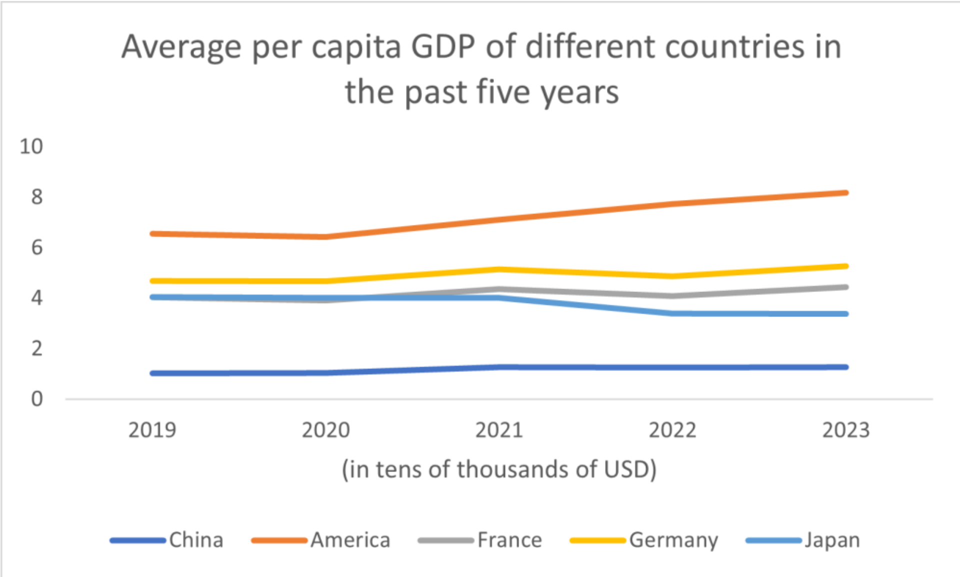


Figure 13

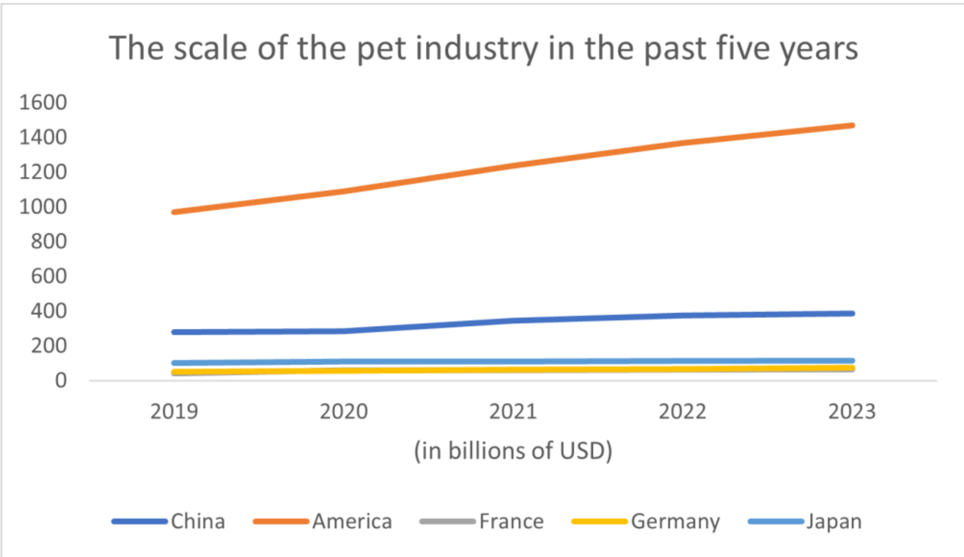


Figure 14

As can be seen from Figure 13, the per capita GDP of these representative countries shows an overall upward trend, which indicates that the global consumption power has increased to a certain extent. From Figure 14, it can be seen that the pet market in these five countries has shown a trend of gradual expansion in the past five years. The correlation between the two trends coincides with the analysis results of the first question.

5.2.3 Analysis and Summary of the Development of the Global Pet Market Over the Past Five Years

As can be seen from Figure 10、Figure 11、Figure 12、Figure 13 and Figure 14 the total number of pets in the world has remained relatively stable in recent years. As for the number of cats, the overall number is increasing, among which the number of cats in China has continued to increase in the past five years, while the number of cats in the other four countries has fluctuated. As for the number of dogs, it can be seen that the total number is showing a downward trend, except for France and Germany, two European countries where the number of dogs is increasing, the number of dogs in other countries has declined. However, the overall number of cats and dogs is huge, and the global per capita GDP is generally rising, so the scale of the pet market is developing upward, especially in the United States, the scale of the pet industry has continued to expand in the past five years.

In general, due to the large base of cats and dogs and the development of per capita GDP, the global pet industry has been developing continuously in the past five years.

5.2.4 Model establishment for the pet food market

The process of establishing the model is the same as the first question, and the result of establishing the model is given directly instead of repeating it.

About America:

Table 8

Model Statistics	
Variable	Model Type
Food consumption	ARIMA(3,1,0)
Per capita GDP	ARIMA(3,0,2)
Pet count	ARIMA(1,1,3)
Pet industry	ARIMA(3,1,0)

About France:

Table 9

Model Statistics	
Variable	Model Type
Food consumption	ARIMA(3,1,0)
Per capita GDP	ARIMA(0,0,0)
Pet count	ARIMA(0,1,0)
Pet industry	ARIMA(0,1,0)

(For detailed models for the five specific countries as well as the global model , please refer to the appendix.)

5.2.5 Forecast for the pet food market over the next three year

Based on the aforementioned models, the predicted results obtained through Python calculations are presented in Table 10:

Table 10 Global Forecasted Pet Food Sales (2024-2026)

Year	Forecasted_Pet_Food_Demands(unit: billion USD)
2024	96.236119
2025	110.371410
2026	119.903496

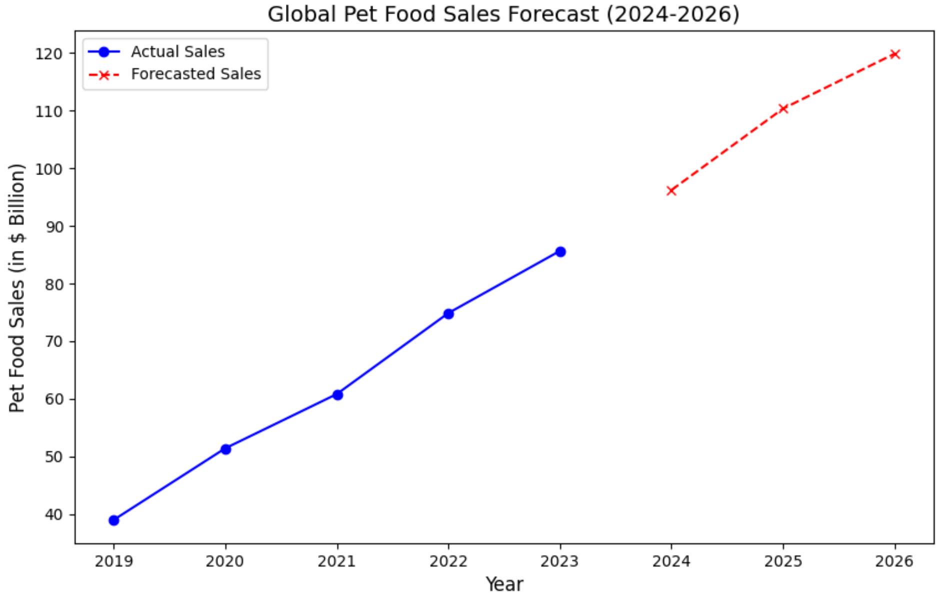


Figure 15

Conclusion It can be seen that the global pet food market demand in the next three years is still showing an upward trend, it can be seen that the demand for pet food in 2024 is nearly 10 billion yuan, and it will exceed 10 billion yuan from 2025, and the demand continues to rise.

Verify Through calculation, it is found that the determination coefficient of the global ARIMA model is 0.949,indicating that the fitting effect is good.

5.3 Establishment and solution of Problem-3 model

5.3.1 Data collection and Preprocessing

Since this question requires analysis based on the global pet food market demand trend and the development situation in China, the data required for this question are: The production value of pet food in China, the number of pet food enterprises in China, the proportion of food types produced in China, the export value of pet food in China, etc., among which the demand for pet food in China and the global demand for pet food have been covered in the previous topic, and will not be repeated here.

5.3.2 Analyze and summarize the development of pet food industry in China

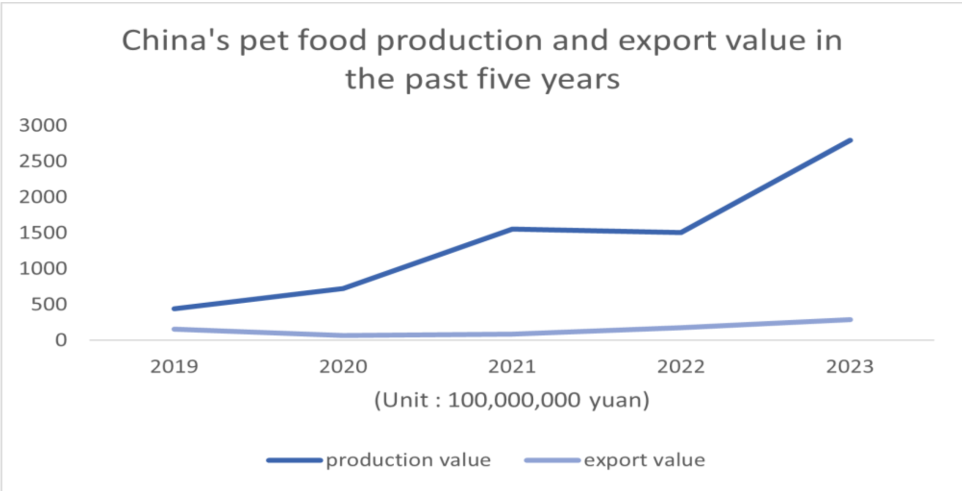


Figure 16

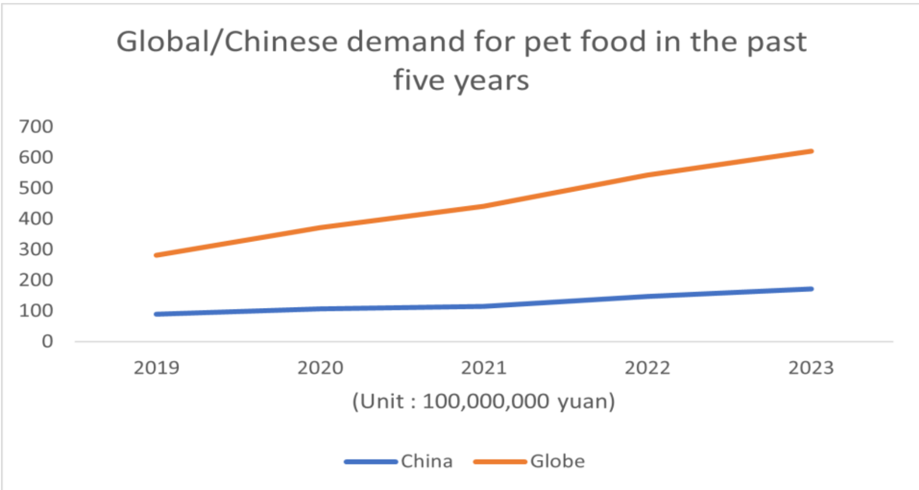


Figure 17

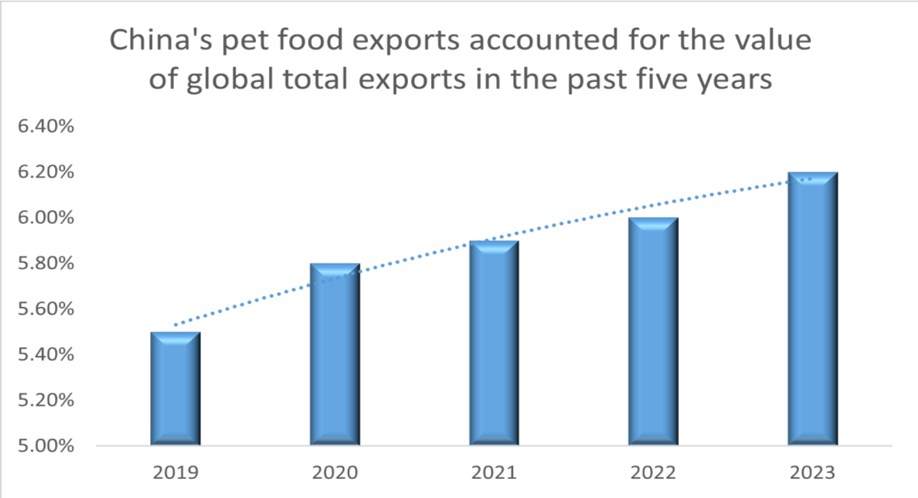


Figure 18

According to Figure 16, China's pet food production and exports have increased significantly over the past five years, with production growing more rapidly. Given the expanding global demand for pet food (as per the second problem), China's production exceeds its exports. Figure 17 shows that China's domestic demand growth is less than global demand growth, indicating China can expand foreign markets while meeting domestic needs. Figure 18 reveals China's increasing export share in the global market, suggesting a promising outlook for China's pet industry.

5.3.3 Forecast China's pet food production and export in the next three years

Table 11

Model Statistics	
Variable	Model Type
Domestic_Production	ARIMA(3,0,3)
Global_Pet_Food_Demand	ARIMA(3,1,0)
China_Export_Share	ARIMA(1,1,0)
China_Export_Value	ARIMA(2,1,3)

The calculation results are in
Table 12：

Table 12

Year	Export Value(unit:billion yuan)	Production Value(unit:billion yuan)
2024	299.10	2818.48
2025	324.76	3056.40

2026	343.53	3294.27

The trend in the next three years is as follows:

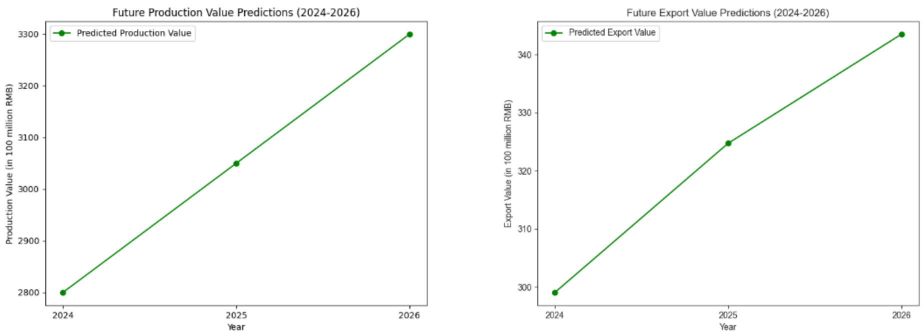


Figure 19

Verify Through calculation, it is found that the determination coefficient of the Export Value and Production Value are 0.90 and 0.92,indicating that the fitting effect is good.

5.4 Establishment and solution of Problem-4 model

5.4.1 Establishment of pet food impact analysis model based on price elasticity of demand

As this topic needs to study and formulate feasible strategies for the sustainable development of China's pet food industry under different economic policies and conduct quantitative analysis of them, this paper intends to formulate strategies based on the degree of changes in the production and export value of China's pet food. In this study, we can introduce the price elasticity of demand in economics, which can well reflect how the demand changes with the change of price, so as to reflect the degree of change in the value of production and export. At the same time, in order to simplify the model, we only study the United States as the country that promulgates economic policies towards China, and the tariff rate is equal to the price change rate of China's pet food industry.

The specific model is shown as follows:

For production:

$$\begin{cases} \Delta Q = -e_d \cdot Q_0 \cdot \dfrac{\Delta P}{P_0} \\ \dfrac{\Delta P}{P_0} = \alpha \end{cases} \tag{10}$$

For export:

$$\begin{cases} \Delta Q' = -e_d' \cdot Q_0' \cdot \dfrac{\Delta P'}{P_0'} \\ \dfrac{\Delta P'}{P_0'} = \alpha \end{cases} \tag{11}$$

Where, ΔQ、$\Delta Q'$represent the changes in the production value and export value of China's pet food industry, Q_0、Q_0'represent the production value and export value of China's pet food industry before tariff adjustment, and e_d、e_d'represent the price elasticity of demand in China and the US pet food market. ΔP、$\Delta P''$ represent the changes in Chinese pet food market prices caused by tariffs, P_0、P_0' represent the Chinese market prices before tariff adjustment, and α is the tariff rate imposed by the United States on China.

After the model is established, this paper intends to conduct research based on 2023, and the collected values of α is 30%, e_d is 0.5, and e_d' is -0.3. After that, this paper conducts a sensitivity analysis with a tariff rate range of ±10%, so as to formulate a feasible development strategy for China's pet food industry.

5.4.2 Solving the impact analysis model of pet food based on price elasticity of demand

Using Python programming drawing:

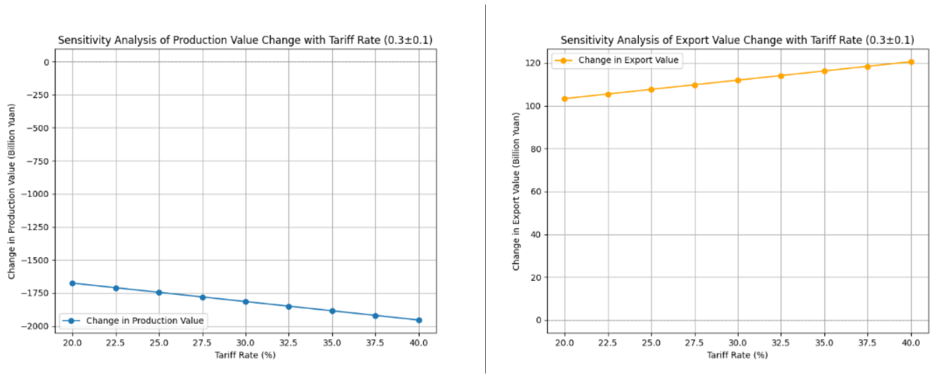


Figure 20

The left chart shows a linear decline, indicating higher tariffs increase production costs and reduce economic efficiency, hampering China's pet food output. The right chart, however, shows the export value rising with tariff increases, reflecting China's aggressive pet food export expansion, potentially at the cost of domestic consumption.

5.4.3 Strategies

- ✧ Negotiate with major trade partners to reduce tariffs on key raw materials, thereby lowering production costs.
- ✧ Consider using sustainable protein sources as alternatives to traditional meat raw materials to reduce carbon footprints and dependence on high-cost imported raw materials.
- ✧ Obtain sustainable certifications for products to make Chinese pet food more attractive in the international market, especially in regions with high environmental concerns such as the European Union and North America.

VI. Evaluation and Extension of the Model

6.1 Advantages of the model

(1) Stable performance: The data involved in this paper have a large time span, regional span and complex data type dimension. The ARIMA model is adopted for prediction, which can perform stably when processing time series data and give better prediction results for different data sets.

(2) Effective prediction: Although the data required in this paper is incomplete, ARIMA model can process non-stationary time series data and convert non-stationary time series into stationary time series through differential processing of data, so as to carry out effective modeling and prediction.

(3) When solving for the value of the dependent variable, use both Random Forest and Decision Tree simultaneously and compare their results to select the better one, in order to achieve more accurate outcomes.

6.2 Disadvantages of the model

(1) Since the main data in this paper are collected by our team on the network, there is no guarantee for the accuracy of the data.

(2) Because it is not possible to collect data from individual countries, the analysis of global trends can only be discussed on behalf of a few countries, which may be biased from the actual situation.

6.3 Model improvement and extension

(1) This paper mainly focuses on the group characteristics of mainstream cats and dogs, for which we can further refine, analyze the market development of different types of cats and dogs, or analyze the market development of other animal types.

(2) The data analyzed in this paper are all discrete data, but the random forest model can also process continuous data and does not need to standardize the data set, which can be used in financial risk control, medical treatment, e-commerce and other fields.

VII. References

[1] KepinskaPacelik, Jagoda,Biel, Wioletta.Insects in Pet Food Industry-Hope or Threat?[J].ANIMALS,2022,12(12).DOI:10.3390/ani12121515.

[2] Charmaraman, Linda,Mueller, Megan K.,Richer, Amanda M..The Role of Pet Companionship in Online and Offline Social Interactions in Adolescence[J].CHILD AND ADOLESCENT SOCIAL WORK JOURNAL,2020,37(06):589-599.DOI:10.1007/s10560-020-00707-y.

[3] Jenny Hu.The Pet Medical Market Is to Accelerate Expansion[J].China's Foreign Trade,2021,(06):47-49.

[4] Li, Zhongqi,Tao, Bilin,Zhan, Mengyao, et al.A comparative study of time series models in predicting COVID-19 cases[J].Chinese Journal of Epidemiology,2021,42(03):421-426.DOI:10.3760/cma.j.cn112338-20201116-01333.

[5] YE Liming,YANG Guixia,Eric VAN RANST, et al.Time-Series Modeling and Prediction of Global Monthly Absolute Temperature for Environmental Decision Making[J].Advances in Atmospheric Sciences,2013,30(02):382-396.

[6] Wang, Lujie,Li, Jiajie,Lin, Yixuan, et al.Establishment and external validation of an early warning model of diabetic peripheral neuropathy based on random forest and logistic regression[J].BMC ENDOCRINE DISORDERS,2024,24(01).DOI:10.1186/s12902-024-01728-9.

[7] Yu, Hong,Sun, Yujing,Ren, Jiaxin, et al.Factors related to suicidal ideation of schizophrenia patients in China: a study based on decision tree and logistic regression model[J].PSYCHOLOGY HEALTH & MEDICINE,2024,29(07):1281-1295.DOI:10.1080/13548506.2023.2301225.

Appendices

Appendix 1

This appendix contains the statistics of the influencing factors from Question 1, as well as the ARIMA model types for different countries and globally, including the preprocessed data.

Variable Name	Sample Size	Maximum Value	Minimum Value	Mean	Standard Deviation	Median	Variance	Kurtosis	Skewness	Coefficient of Variation
Number of pet dogs (ten thousand)	5	5503	5119	5289.6	167.1669824	5222	27944.8	-2.410549471	0.503754715	0.031602953
Number of pet cats (ten thousand)	5	6980	4412	5719.2	1084.882113	5806	1176969.2	-2.196143872	-0.102395137	0.189691235
GDP per capita (USD)	5	12662	10143	11688.8	1293.697685	12614	1673653.7	-3.192627737	-0.631460433	0.1106784
Urbanization rate	5	66.16	62.71	64.54	1.312497619	64.72	1.72265	-0.209000486	-0.337487292	0.020336189
Aging population ratio	5	0.154	0.126	0.1412	0.01112205	0.142	0.0001237	-1.023435973	-0.356738519	0.078768061

Single rate	5	0.246	0.225	0.2366	0.008294577	0.238	6.88E-05	-0.812897174	-0.487324596	0.035057382
Number of pet businesses (ten thousand)	5	236	50.2	151.84	86.867876699	190	7546.028	-3.000102444	-0.473484614	0.572101401
Number of households with pets (ten thousand)	5	10565	5989	8378.2	2071.989069	9168	4293138.7	-2.830397801	-0.371205018	0.247307186

Year	Cat_Count	Dog_Count	GDP_Per_Capita	Urbanization_Rate	Population_Aging	Single_Rate	Pet_Industry_Company	Family_Number
2019	-1.06762432	1.13047418	-1.131579937	-1.078536197	-1.093769303	-1.110246308	-1.107657626	-1.090709596
2020	-0.751797508	-0.390566952	-0.969912591	-0.435472917	-0.498342661	-0.492756492	-0.956384584	-0.948345664
2021	0.076370576	0.792211088	0.714630655	0.129932297	0.080544353	0.168839739	0.428033887	0.362432768
2022	0.716797166	-0.915103301	0.7420836	0.470537848	0.659431366	0.609903892	0.720675189	0.658399888

2023	1.028412953	-0.659120421	0.712342909	0.982808596	0.990223945	0.962755216	0.900762145	0.94500096

	Year	Food_Consumption	GDP_Per_Capita	Pet_Count	Pet_Industry	Pet_Food_Sales
Global	2019	1.543413441	1.944995153	1.924667839	1.170830327	0.700316419
Global	2020	1.761135357	1.905128767	1.684621413	1.391035262	1.201780228
Global	2021	1.872439578	2.136645601	1.924667839	1.687638112	1.58181153
Global	2022	1.959786244	2.093803201	1.922227313	1.819868715	1.893784148
Global	2023	1.959786244	2.136645601	1.920636576	1.819868715	1.893784148

Year	Domestic_Production	Global_Pet_Food_Demand	China_Export_Share	China_Export_Value
2019	-1.134311774	-1.037231714	-1.057633442	-0.060981099
2020	-0.95846556	-0.541495407	-0.252818349	-0.891384733
2021	0.719579101	-0.07236059	0.082521273	-0.746204896
2022	0.745312693	0.628076593	0.417860895	0.198715592
2023	0.703495606	1.057549756	0.95440429	1.099831819

About China:

Model Statistics	
Variable	Model Type
Food consumption	ARIMA(3,1,0)
Per capita GDP	ARIMA(3,0,3)
Pet count	ARIMA(0,1,2)
Pet industry	ARIMA(3,1,0)

About Germany:

Model Statistics	
Variable	Model Type
Food consumption	ARIMA(3,1,0)
Per capita GDP	ARIMA(3,1,0)
Pet count	ARIMA(3,1,0)
Pet industry	ARIMA(1,1,0)

About Japan:

Model Statistics	
Variable	Model Type
Food consumption	ARIMA(3,1,0)
Per capita GDP	ARIMA(3,0,2)
Pet count	ARIMA(0,1,0)
Pet industry	ARIMA(0,1,0)

About the world:

Model Statistics	
Variable	Model Type

Food consumption	ARIMA(3,1,0)
Per capita GDP	ARIMA(3,0,2)
Pet count	ARIMA(0,1,2)
Pet industry	ARIMA(3,1,0)

Appendix 2

This code is written in Python and combines time series modeling (ARIMA) with machine learning methods (Random Forest Regression) to predict the future trends of pet cat and pet dog populations. It also uses hyperparameter optimization and model evaluation to ensure the accuracy of the predictions.

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

warnings.filterwarnings('ignore')
from statsmodels.tsa.arima.model import ARIMA
from scipy.stats import spearmanr

# Prepare data
data = {
    'Year': [2019, 2020, 2021, 2022, 2023],
    'Cat_Count': [4412, 4862, 5806, 6536, 6980],  # Number of pet cats (in
10,000s)
    'Dog_Count': [5503, 5222, 5429, 5119, 5175],  # Number of pet dogs (in
```

```
10,000s)
            'GDP_Per_Capita': [10143, 10408, 12617, 12662, 12614],   # GDP per
capita (in USD)
            'Urbanization_Rate': [62.71, 63.89, 64.72, 65.22, 66.16],   # Urbanization
rate (%)
            'Population_Aging': [0.126, 0.135, 0.142, 0.149, 0.154],   # Aging
population ratio
            'Single_Rate': [0.225, 0.232, 0.238, 0.242, 0.246],   # Single rate
            'Pet_Industry_Company': [50.2, 67, 190, 216, 236],   # Number of pet
industry companies
            'Family_Number': [5989, 6369, 9168, 9800, 10565],   # Number of pet-
owning households
      }

      # Create DataFrame
      df = pd.DataFrame(data)

      # Use ARIMA model to forecast pet numbers
      # ARIMA model setup
      cat_arima_model = ARIMA(df['Cat_Count'], order=(1, 1, 1))
      dog_arima_model = ARIMA(df['Dog_Count'], order=(1, 1, 1))

      # Train ARIMA model
      cat_arima_result = cat_arima_model.fit()
      dog_arima_result = dog_arima_model.fit()

      # Forecast for the next three years
      cat_arima_forecast = cat_arima_result.forecast(steps=3)
      dog_arima_forecast = dog_arima_result.forecast(steps=3)

      # Data fusion, combining ARIMA and Random Forest predictions
      # Currently, averaging the two prediction results
      future_years = [2024, 2025, 2026]
```

```
    # Use previously trained Random Forest models to predict data
    # Feature variables and target variables
    X_cat = df[['Year', 'GDP_Per_Capita', 'Urbanization_Rate', 'Population_Aging',
                    'Single_Rate', 'Pet_Industry_Company', 'Family_Number']]
    y_cat = df['Cat_Count']

    X_dog = df[['Year', 'GDP_Per_Capita', 'Urbanization_Rate',
'Population_Aging',
                    'Single_Rate', 'Pet_Industry_Company', 'Family_Number']]
    y_dog = df['Dog_Count']

    # Set hyperparameter grid for Random Forest model
    param_grid = {
        'n_estimators': [25, 50, 75, 100, 200],  # Number of trees in the forest
        'max_depth': [4, 5, 6, 10, 15, None],  # Maximum depth of the tree
        'min_samples_split': [2],  # Minimum number of samples required to split
a node
        'min_samples_leaf': [1],  # Minimum number of samples required at a leaf
node
        'max_features': ['sqrt', 'log2']  # Maximum number of features to consider
in each tree
    }

    # Create Random Forest model
    cat_model = RandomForestRegressor(random_state=42)
    dog_model = RandomForestRegressor(random_state=42)

    # Use GridSearchCV for hyperparameter optimization (pet cat model)
    cat_grid_search = GridSearchCV(estimator=cat_model,
param_grid=param_grid, cv=3, scoring='neg_mean_squared_error',
                                   verbose=0, n_jobs=1)
    cat_grid_search.fit(X_cat, y_cat)
    best_cat_model = cat_grid_search.best_estimator_
```

```
# Use GridSearchCV for hyperparameter optimization (pet dog model)
dog_grid_search = GridSearchCV(estimator=dog_model,
param_grid=param_grid, cv=3, scoring='neg_mean_squared_error',
                                 verbose=0, n_jobs=1)
dog_grid_search.fit(X_dog, y_dog)
best_dog_model = dog_grid_search.best_estimator_

# Output best hyperparameters
print("Best hyperparameters - Cat model:")
print(f"Best hyperparameters - Cat model: {cat_grid_search.best_params_}")
print("Best hyperparameters - Dog model:")
print(f"Best hyperparameters - Dog model: {dog_grid_search.best_params_}")

# Predict future feature values (2024-2026)
from sklearn.linear_model import LinearRegression



def predict_future_features(df, feature_names, years):
    predictions = {}
    for feature in feature_names:
        X = df[['Year']]
        y = df[feature]
        model = LinearRegression()
        model.fit(X, y)
        future_values = model.predict(pd.DataFrame({'Year': years}))
        predictions[feature] = future_values
    return predictions



# Predict features
cat_feature_names = ['GDP_Per_Capita', 'Urbanization_Rate',
'Population_Aging',
                     'Single_Rate','Pet_Industry_Company',
'Family_Number']
```

```
    dog_feature_names = ['GDP_Per_Capita', 'Urbanization_Rate',
'Population_Aging',
                          'Single_Rate','Pet_Industry_Company',
'Family_Number']

    cat_future_features = predict_future_features(df, cat_feature_names,
future_years)
    dog_future_features = predict_future_features(df, dog_feature_names,
future_years)

    # Create future feature DataFrame
    future_cat_df = pd.DataFrame({
        'Year': future_years,
        'GDP_Per_Capita': cat_future_features['GDP_Per_Capita'],
        'Urbanization_Rate': cat_future_features['Urbanization_Rate'],
        'Population_Aging': cat_future_features['Population_Aging'],
        'Single_Rate': cat_future_features['Single_Rate'],
        'Pet_Industry_Company': cat_future_features['Pet_Industry_Company'],
        'Family_Number': cat_future_features['Family_Number']
    })

    future_dog_df = pd.DataFrame({
        'Year': future_years,
        'GDP_Per_Capita': dog_future_features['GDP_Per_Capita'],
        'Urbanization_Rate': dog_future_features['Urbanization_Rate'],
        'Population_Aging': dog_future_features['Population_Aging'],
        'Single_Rate': dog_future_features['Single_Rate'],
        'Pet_Industry_Company': dog_future_features['Pet_Industry_Company'],
        'Family_Number': dog_future_features['Family_Number']
    })

    # Use the best model to make predictions
    cat_rf_predictions = best_cat_model.predict(future_cat_df)
    dog_rf_predictions = best_dog_model.predict(future_dog_df)
```

```
    # Combine ARIMA and Random Forest predictions
    cat_final_predictions = (cat_arima_forecast + cat_rf_predictions) / 2
    dog_final_predictions = (dog_arima_forecast + dog_rf_predictions) / 2

    # Evaluate model performance
    cat_train_pred = best_cat_model.predict(X_cat)
    dog_train_pred = best_dog_model.predict(X_dog)

    cat_mse = mean_squared_error(y_cat, cat_train_pred)
    cat_mae = mean_absolute_error(y_cat, cat_train_pred)
    cat_r2 = r2_score(y_cat, cat_train_pred)

    dog_mse = mean_squared_error(y_dog, dog_train_pred)
    dog_mae = mean_absolute_error(y_dog, dog_train_pred)
    dog_r2 = r2_score(y_dog, dog_train_pred)

    print("Model evaluation metrics:")
    print(f"Cat model - Mean Squared Error (MSE): {cat_mse:.2f}, Mean Absolute
Error (MAE): {cat_mae:.2f}, R²: {cat_r2:.2f}")
    print(f"Dog model - Mean Squared Error (MSE): {dog_mse:.2f}, Mean
Absolute Error (MAE): {dog_mae:.2f}, R²: {dog_r2:.2f}")

    # Output future predictions
    print("Future pet number predictions:")
    for year, cat_pred, dog_pred in zip(future_years, cat_final_predictions,
dog_final_predictions):
        print(f"Year {year}: Predicted cat count: {cat_pred:.2f} (10,000s),
Predicted dog count: {dog_pred:.2f} (10,000s)")

    sns.set_style("ticks")  # Choose a style without grid lines
    # Visualize the results
    sns.set(style="whitegrid")
```

```
sns.set_style("ticks")  # Choose a style without grid lines

# Actual vs predicted comparison (training set) Set the x-axis to integer years
def plot_actual_vs_predicted(years, actual, predicted, title):
    plt.figure(figsize=(8, 6))
    plt.plot(years, actual, label='Actual', marker='o')
    plt.plot(years, predicted, label='Predicted', marker='s')
    plt.title(title)
    plt.xlabel('Year')
    plt.ylabel('Number of Pets (in 10,000s)')
    plt.xticks(ticks=years)  # Only display integer years
    plt.legend()
    plt.show()



plot_actual_vs_predicted(df['Year'], y_cat, cat_train_pred, 'Cat Count: Actual vs Predicted')
plot_actual_vs_predicted(df['Year'], y_dog, dog_train_pred, 'Dog Count: Actual vs Predicted')

sns.set_style("ticks")  # Choose a style without grid lines

# Future predictions visualization Adjust the x-axis for future prediction graph
def plot_future_predictions(years, predictions, title):
    plt.figure(figsize=(8, 6))
    plt.plot(years, predictions, label='Predicted', marker='o')
    plt.title(title)
    plt.xlabel('Year')
    plt.ylabel('Number of Pets (in 10,000s)')
    plt.xticks(ticks=years)  # Only display integer years
    plt.legend()
    plt.show()
```

```
    plot_future_predictions(future_years, cat_final_predictions, 'Future Cat Count
Predictions')
    plot_future_predictions(future_years, dog_final_predictions, 'Future Dog
Count Predictions')



    # Visualize Spearman correlation matrix
    def plot_spearman_correlation(df, target_columns, feature_columns):
        # Build the correlation matrix
        correlation_matrix = np.zeros((len(target_columns),
len(feature_columns)))

        for i, target in enumerate(target_columns):
            for j, feature in enumerate(feature_columns):
                correlation, _ = spearmanr(df[target], df[feature])
                correlation_matrix[i, j] = correlation

        # Create a DataFrame for visualization
        correlation_df = pd.DataFrame(correlation_matrix, index=target_columns,
columns=feature_columns)

        sns.set_style("ticks")  # Choose a style without grid lines

        # Draw heatmap
        plt.figure(figsize=(10, 6))
        sns.heatmap(correlation_df, annot=True, cmap="coolwarm",
linewidths=0.5, center=0, vmin=-1, vmax=1)
        plt.title("Spearman Correlation Heatmap")
        plt.show()



    # Define target variables and feature variables
    target_columns = ['Cat_Count', 'Dog_Count']
    feature_columns = ['GDP_Per_Capita', 'Urbanization_Rate',
```

```
'Population_Aging', 'Single_Rate', 'Pet_Industry_Company', 'Family_Number']

    sns.set_style("ticks")  # Choose a style without grid lines

    # Call the plotting function
    plot_spearman_correlation(df, target_columns, feature_columns)
```

Appendix 3

This code is written in Python and combines the ARIMA time series model with the Decision Tree Regression model. It uses GridSearchCV for hyperparameter optimization and addresses the problem of predicting pet cat and pet dog populations.

```
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA
import itertools
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score

# Define data for each country and global data
countries_data = {
    'China': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [160, 137, 151, 166, 186],  # Food consumption (in billion)
        'GDP_Per_Capita': [1.0143, 1.0408, 1.2617, 1.2662, 1.2614],  # GDP per capita (in 10,000 USD)
        'Pet_Count': [9915, 10084, 11235, 11655, 12155],  # Pet count (in 10,000s)
        'Pet_Industry': [279.31, 284.97, 343.62, 373.43, 385.43],  # Pet industry value (in billion USD)
        'Pet_Food_Sales': [12.4, 14.7, 15.8, 20.4, 23.7]  # Pet food sales (in billion USD)
```

```
    },
    'USA': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [2862, 3113, 3399, 3582, 3828],
        'GDP_Per_Capita': [6.5548, 6.4317, 7.1055, 7.7246, 8.1695],
        'Pet_Count': [18390, 15000, 18390, 16350, 15390],
        'Pet_Industry': [971, 1089, 1236, 1368, 1470],
        'Pet_Food_Sales': [20.4, 27.1, 29.4, 35.1, 38.4]
    },
    'France': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [2243, 2534, 2746, 2854, 3166],
        'GDP_Per_Capita': [4.0494, 3.9179, 4.3671, 4.0886, 4.446],
        'Pet_Count': [2040, 2265, 2260, 2250, 2650],
        'Pet_Industry': [40.63, 60.42, 60, 62.7, 64.3],
        'Pet_Food_Sales': [2.64, 3.74, 5.47, 6.1, 7.47]
    },
    'Germany': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [3412, 3670, 3846, 4110, 4351],
        'GDP_Per_Capita': [4.6805, 4.6749, 5.1426, 4.8717, 5.2745],
        'Pet_Count': [2480, 2640, 2700, 2580, 2620],
        'Pet_Industry': [52.67, 57.44, 62.53, 67.58, 73.89],
        'Pet_Food_Sales': [2.26, 3.45, 6.4, 7.41, 8.64]
    },
    'Japan': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [611, 636, 358, 373, 693],
        'GDP_Per_Capita': [4.0415, 4.004, 4.0058, 3.4017, 3.3834],
        'Pet_Count': [1652, 1597, 1606, 1589, 1590],
        'Pet_Industry': [99.77, 109.07, 110.92, 113.34, 116.15],
        'Pet_Food_Sales': [1.26, 2.4, 3.74, 5.86, 7.45]
    },
    'Global': {
```

```
            'Year': [2019, 2020, 2021, 2022, 2023],
            'Food_Consumption': [9288, 10090, 10500, 11085, 12224],
            'GDP_Per_Capita': [20.3405, 20.0693, 21.8827, 21.3528, 22.5348],
            'Pet_Count': [34477, 31586, 36191, 34424, 34405],
            'Pet_Industry': [1443.38, 1600.9, 1813.07, 1985.05, 2109.77],
            'Pet_Food_Sales': [38.96, 51.39, 60.81, 74.87, 85.66]
        }
}

# Forecast for each country and globally, and plot results
forecast_years = [2024, 2025, 2026]

for country, data in countries_data.items():
    df = pd.DataFrame(data)
    y = df['Pet_Food_Sales']

    # Use automated grid search to optimize ARIMA model parameters
    p = d = q = range(0, 3)
    pdq = list(itertools.product(p, d, q))

    best_aic = float('inf')
    best_pdq = None
    best_model = None

    for param in pdq:
        try:
            temp_model = ARIMA(y, order=param)
            temp_result = temp_model.fit()
            if temp_result.aic < best_aic:
                best_aic = temp_result.aic
                best_pdq = param
                best_model = temp_result
        except:
            continue
```

```
    # Fit ARIMA model with best parameters
    print(f"Best ARIMA model for {country}: order={best_pdq} with AIC={best_aic:.2f}")

    # Calculate R²  for the fitted model
    y_pred = best_model.fittedvalues   # Get fitted values
    y_actual = y[best_pdq[1]:]   # If there is differencing, skip the initial differenced part
    y_pred = y_pred[best_pdq[1]:]   # Ignore differenced parts of the fitted values as well
    r2 = r2_score(y_actual, y_pred)   # Calculate R²  value
    print(f"{country} R²  value for the fitted ARIMA model: {r2:.4f}")

    # Forecast pet food sales for the next three years
    forecast = best_model.forecast(steps=3)

    # Combine historical data and forecast data
    combined_years = list(df['Year']) + forecast_years
    combined_sales = list(y) + list(forecast)

    # Plot the forecast
    plt.figure(figsize=(10, 6))
    plt.plot(df['Year'], y, label='Actual Sales', marker='o', color='b')
    plt.plot(forecast_years, forecast, label='Forecasted Sales', marker='x', linestyle='--', color='r')
    plt.title(f'{country} Pet Food Sales Forecast (2024-2026)', fontsize=14)
    plt.xlabel('Year', fontsize=12)
    plt.ylabel('Pet Food Sales (in $ Billion)', fontsize=12)
    plt.xticks(ticks=combined_years, labels=combined_years)
    plt.legend()
    plt.grid(False)   # Set grid=False to remove background lines
    plt.show()
```

```
    # Output forecast results
    forecast_results = pd.DataFrame({
        'Year': forecast_years,
        'Forecasted_Pet_Food_Sales': forecast
    })
    print(f"\n{country} Forecasted Pet Food Sales (2024-2026):")
    print(forecast_results)
```

Appendix 4

This code is written in Python and uses the ARIMA model to perform fitting analysis on multiple time series variables. It selects the best ARIMA model parameters, which serve as the preparatory conditions for the ARIMA models in Questions 1 and 2.

```
import warnings

import pandas as pd
from statsmodels.tsa.arima.model import ARIMA

warnings.filterwarnings('ignore')

# Data preparation
data = {
    'Year': [2019, 2020, 2021, 2022, 2023],
    'Food_Consumption': [9288, 10090, 10500, 11085, 12224],   # Global food consumption (in billions)
    'GDP_Per_Capita': [20.3405, 20.0693, 21.8827, 21.3528, 22.5348],   # GDP per capita (in 10,000 USD, global)
    'Pet_Count': [34477, 31586, 36191, 34424, 34405],   # Pet count (in 10,000s, global)
    'Pet_Industry': [1443.38, 1600.9, 1813.07, 1985.05, 2109.77]   # Pet industry size (in billion USD, global)
}
```

```
df = pd.DataFrame(data)

# Define function: Perform ARIMA model fitting and parameter selection for each
variable
def arima_model_analysis(series, variable_name):
    print(f"\n{'-'*50}")
    print(f"Analyzing ARIMA Model for {variable_name}...\n")
    results = []

    # Define ARIMA parameter range
    p_values = range(0, 4)
    d_values = range(0, 2)
    q_values = range(0, 4)

    for p in p_values:
        for d in d_values:
            for q in q_values:
                try:
                    # Fit ARIMA model
                    model = ARIMA(series, order=(p, d, q))
                    result = model.fit()

                    # Record AIC/BIC values
                    results.append({
                        'p': p,
                        'd': d,
                        'q': q,
                        'AIC': result.aic,
                        'BIC': result.bic
                    })
                except:
                    continue
```

```
    # Convert to DataFrame and sort by AIC
    results_df                                                              =
pd.DataFrame(results).sort_values(by='AIC').reset_index(drop=True)

    # Output the best parameter information
    print("Top 5 ARIMA Models (based on AIC):")
    print(results_df.head())

    return results_df

# Perform ARIMA analysis for each individual variable
variables = ['Food_Consumption', 'GDP_Per_Capita', 'Pet_Count', 'Pet_Industry']
analysis_results = {}

for var in variables:
    analysis_results[var] = arima_model_analysis(df[var], var)
```

Appendix 5

This code is written in Python. It uses the ARIMA model to perform time series forecasting on pet food sales data for each country and globally, and optimizes the model's hyperparameters. Ultimately, it generates pet food sales forecasts for each country and globally for the next three years.

```
import warnings

import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

```
from sklearn.model_selection import GridSearchCV

warnings.filterwarnings('ignore')
from statsmodels.tsa.arima.model import ARIMA
import itertools

# Define the data for each country and global
countries_data = {
    'China': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [160, 137, 151, 166, 186],    # Food consumption (in
billion)
        'GDP_Per_Capita': [1.0143, 1.0408, 1.2617, 1.2662, 1.2614],    # GDP per
capita (in 10,000 USD)
        'Pet_Count': [9915, 10084, 11235, 11655, 12155],    # Pet count (in
10,000s)
        'Pet_Industry': [279.31, 284.97, 343.62, 373.43, 385.43],    # Pet industry
value (in billion USD)
        'Pet_Food_Sales': [12.4, 14.7, 15.8, 20.4, 23.7]    # Pet food sales (in billion
USD)
    },
    'USA': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [2862, 3113, 3399, 3582, 3828],
        'GDP_Per_Capita': [6.5548, 6.4317, 7.1055, 7.7246, 8.1695],
        'Pet_Count': [18390, 15000, 18390, 16350, 15390],
        'Pet_Industry': [971, 1089, 1236, 1368, 1470],
        'Pet_Food_Sales': [20.4, 27.1, 29.4, 35.1, 38.4]
    },
    'France': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [2243, 2534, 2746, 2854, 3166],
        'GDP_Per_Capita': [4.0494, 3.9179, 4.3671, 4.0886, 4.446],
        'Pet_Count': [2040, 2265, 2260, 2250, 2650],
```

```
        'Pet_Industry': [40.63, 60.42, 60, 62.7, 64.3],
        'Pet_Food_Sales': [2.64, 3.74, 5.47, 6.1, 7.47]
    },
    'Germany': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [3412, 3670, 3846, 4110, 4351],
        'GDP_Per_Capita': [4.6805, 4.6749, 5.1426, 4.8717, 5.2745],
        'Pet_Count': [2480, 2640, 2700, 2580, 2620],
        'Pet_Industry': [52.67, 57.44, 62.53, 67.58, 73.89],
        'Pet_Food_Sales': [2.26, 3.45, 6.4, 7.41, 8.64]
    },
    'Japan': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [611, 636, 358, 373, 693],
        'GDP_Per_Capita': [4.0415, 4.004, 4.0058, 3.4017, 3.3834],
        'Pet_Count': [1652, 1597, 1606, 1589, 1590],
        'Pet_Industry': [99.77, 109.07, 110.92, 113.34, 116.15],
        'Pet_Food_Sales': [1.26, 2.4, 3.74, 5.86, 7.45]
    },
    'Global': {
        'Year': [2019, 2020, 2021, 2022, 2023],
        'Food_Consumption': [9288, 10090, 10500, 11085, 12224],
        'GDP_Per_Capita': [20.3405, 20.0693, 21.8827, 21.3528, 22.5348],
        'Pet_Count': [34477, 31586, 36191, 34424, 34405],
        'Pet_Industry': [1443.38, 1600.9, 1813.07, 1985.05, 2109.77],
        'Pet_Food_Sales': [38.96, 51.39, 60.81, 74.87, 85.66]
    }
}

# Forecast for each country and globally, and plot results
forecast_years = [2024, 2025, 2026]

for country, data in countries_data.items():
    df = pd.DataFrame(data)
```

```
    y = df['Pet_Food_Sales']

    # Use automated grid search to optimize ARIMA model parameters
    p = d = q = range(0, 3)
    pdq = list(itertools.product(p, d, q))

    best_aic = float('inf')
    best_pdq = None
    best_model = None

    for param in pdq:
        try:
            temp_model = ARIMA(y, order=param)
            temp_result = temp_model.fit()
            if temp_result.aic < best_aic:
                best_aic = temp_result.aic
                best_pdq = param
                best_model = temp_result
        except:
            continue

    # Fit ARIMA model with best parameters
    print(f"Best ARIMA model for {country}: order={best_pdq} with AIC={best_aic:.2f}")

    # Calculate R² for the fitted model
    y_pred = best_model.fittedvalues   # Get fitted values
    y_actual = y[best_pdq[1]:]   # If there is differencing, skip the initial differenced part
    y_pred = y_pred[best_pdq[1]:]   # Ignore differenced parts of the fitted values as well
    r2 = r2_score(y_actual, y_pred)   # Calculate R² value
    print(f"{country} R² value for the fitted ARIMA model: {r2:.4f}")
```

```
# Forecast pet food sales for the next three years
forecast = best_model.forecast(steps=3)

# Combine historical data and forecast data
combined_years = list(df['Year']) + forecast_years
combined_sales = list(y) + list(forecast)

# Plot the forecast
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], y, label='Actual Sales', marker='o', color='b')
plt.plot(forecast_years, forecast, label='Forecasted Sales', marker='x',
linestyle='--', color='r')
plt.title(f'{country} Pet Food Sales Forecast (2024-2026)', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Pet Food Sales (in $ Billion)', fontsize=12)
plt.xticks(ticks=combined_years, labels=combined_years)
plt.legend()
plt.grid(False)  # Set grid=False to remove background lines
plt.show()

# Output forecast results
forecast_results = pd.DataFrame({
    'Year': forecast_years,
    'Forecasted_Pet_Food_Sales': forecast
})
print(f"\n{country} Forecasted Pet Food Sales (2024-2026):")
print(forecast_results)
```

Appendix 6

The following code is written in Python. It combines the ARIMA model and the Random Forest Regression model to predict China's pet food export values for the next three years. The code also evaluates the models and visualizes the results.

```
# -*- coding: utf-8 -*-
import warnings

import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import GridSearchCV

warnings.filterwarnings('ignore')
from statsmodels.tsa.arima.model import ARIMA

# ================================================
# 1. Data loading and preprocessing
# ================================================

# Prepare data
data = {
    'Year': [2019, 2020, 2021, 2022, 2023],
    'Domestic_Production': [14.28, 14.69, 17.82, 17.88, 17.79],  # Domestic production volume
    'Global_Pet_Food_Demand': [282.2, 372.35, 440.6, 542.5, 620.6],  # Global pet food demand (in hundred million RMB)
    'China_Export_Share': [5.5, 5.8, 5.9, 6.0, 6.2],  # China's share of global pet food exports (%)
    'China_Export_Value': [154.1, 71, 88.4, 179, 287]  # China's pet food export value (in hundred million RMB)
}

# Create DataFrame
df = pd.DataFrame(data)
```

```
# ==================================================
# 2. Use ARIMA model to forecast features
# ==================================================

# Use ARIMA model to forecast future features
def predict_future_features_arima(df, feature_names, years):
    predictions = {}
    for feature in feature_names:
        # Use ARIMA model to make predictions
        model = ARIMA(df[feature], order=(1, 1, 1))   # Using (1, 1, 1) as an
example, can find the best parameters through grid search
        model_fit = model.fit()
        future_values = model_fit.forecast(steps=len(years))
        predictions[feature] = future_values
    return predictions

# Define future years
future_years = [2024, 2025, 2026]

# Feature names to forecast
feature_names = ['Domestic_Production', 'Global_Pet_Food_Demand',
'China_Export_Share']

# Use ARIMA model to forecast future features
future_features_arima = predict_future_features_arima(df, feature_names,
future_years)

# Create a DataFrame for future features
future_df_arima = pd.DataFrame({
    'Year': future_years,
    'Domestic_Production': future_features_arima['Domestic_Production'],
    'Global_Pet_Food_Demand':
future_features_arima['Global_Pet_Food_Demand'],
    'China_Export_Share': future_features_arima['China_Export_Share']
```

```
})

# ==============================================
# 3. Use ARIMA and Random Forest regression to fuse the forecast of pet food export value
# ==============================================

# Use ARIMA model to forecast export value
export_arima_model = ARIMA(df['China_Export_Value'], order=(1, 1, 1))
export_arima_result = export_arima_model.fit()
export_arima_forecast = export_arima_result.forecast(steps=3)

# Feature variables and target variable
X = df[['Domestic_Production', 'Global_Pet_Food_Demand', 'China_Export_Share']]
y = df['China_Export_Value']

# Set hyperparameter grid for Random Forest model
param_grid = {
    'n_estimators': [25, 50, 75, 100, 200],   # Number of trees in the forest
    'max_depth': [4, 5, 6, 10, 15, None],   # Maximum depth of the tree
    'min_samples_split': [2],   # Minimum number of samples required to split a node
    'min_samples_leaf': [1],   # Minimum number of samples required at a leaf node
    'max_features': ['sqrt', 'log2']   # Maximum number of features to consider in each tree
}

# Create Random Forest model
rf_model = RandomForestRegressor(random_state=42)

# Use GridSearchCV for hyperparameter optimization
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=3,
```

```
scoring='neg_mean_squared_error',
                                    verbose=0, n_jobs=1)
grid_search.fit(X, y)
best_rf_model = grid_search.best_estimator_

# Output the best hyperparameters
print("Best hyperparameters - Export value model:")
print(f"Best hyperparameters: {grid_search.best_params_}")

# Use the best model for future predictions
rf_predictions = best_rf_model.predict(future_df_arima[feature_names])

# Combine ARIMA and Random Forest predictions
final_predictions = (export_arima_forecast + rf_predictions) / 2

# ==================================================
# 4. Evaluate model performance and visualization
# ==================================================

# Evaluate model performance
train_pred = best_rf_model.predict(X)

mse = mean_squared_error(y, train_pred)
mae = mean_absolute_error(y, train_pred)
r2 = r2_score(y, train_pred)

print("Model evaluation metrics:")
print(f"Export value model - Mean Squared Error (MSE): {mse:.2f}, Mean Absolute
Error (MAE): {mae:.2f}, R²: {r2:.2f}")

# Output future prediction results
print("Future pet food export value predictions for China:")
for year, pred in zip(future_years, final_predictions):
    print(f"Year {year}: Predicted export value: {pred:.2f} hundred million RMB")
```

```
# Visualize results
sns.set_style("ticks")    # Select a style without grid lines

# Actual vs predicted comparison (training set)
plt.figure(figsize=(8, 6))
plt.plot(df['Year'], y, label='Actual', marker='o', color='b')
plt.plot(df['Year'], train_pred, label='Predicted', marker='s', color='r')
plt.title('China Export Value: Actual vs Predicted')
plt.xlabel('Year')
plt.ylabel('Export Value (in 100 million RMB)')
plt.legend()
plt.grid(False)    # Remove grid background
plt.show()

# Visualize future prediction results
plt.figure(figsize=(8, 6))
plt.plot(future_years, final_predictions, label='Predicted Export Value', marker='o',
color='g')
plt.title('Future Export Value Predictions (2024-2026)')
plt.xlabel('Year')
plt.ylabel('Export Value (in 100 million RMB)')
plt.legend()
plt.grid(False)    # Remove grid background
plt.show()
```