Analysis and Forecast of the Pet Industry through Mathematical Models

Abstract

The pet industry has witnessed significant growth globally, driven by increasing disposable income and the rising popularity of pet companionship. This paper explores the development and forecasting of the pet industry using mathematical models, with a focus on China's market trends and global demands. Through the application of ARIMA, polynomial interpolation, and multiple linear regression, this study analyzes the past five years of the pet industry in China, forecasting its development over the next three years. Additionally, global trends and the impact of economic factors, such as tariffs, on the pet food industry are considered. The results provide valuable insights for strategic recommendations and sustainable development in the pet industry.

Figure 1 British short hair cat and golden retriever stock photo [1]

Keywords: Pet Industry, Mathematical Models, ARIMA, Polynomial Interpolation, Multiple Linear Regression, Forecasting, Market Analysis, Economic Factors, Pet Food, Sustainable Development

Contents

I、 Problem Restatement

This problem requires analyzing and forecasting the development trends of the pet industry in China and globally, with a focus on the pet food market. The analysis should combine historical data and additional data collected, building mathematical models for prediction. The specific tasks are as follows:

1.1 Question 1: Development of China's Pet Industry

1.1.1 Objective

Analyze the development of China's pet industry over the past five years by different pet types (e.g., cats, dogs, etc.).

1.1.2 Factors

Look into economic factors, societal trends, and market changes (e.g., income growth, changes in pet ownership behavior, product innovations).

1.1.3 Modeling Task

Develop a mathematical model to predict the growth and development of China's pet industry for the next three years.

1.2 Question 2: Global Pet Industry Development

1.2.1 Objective

Analyze the development of the global pet industry, focusing on different regions (e.g., Europe, America, China).

1.2.2 Factors

Global pet food demand and market characteristics by pet type.

1.2.3 Modeling Task

Create a model to forecast the global demand for pet food over the next three years, based on data provided.

1.3 Question 3: China’s Pet Food Industry

1.3.1 Objective

Analyze the development of China’s pet food industry, specifically its production and export values.

1.3.2 Factors

The trend in China’s pet food production and exports, factoring in the global demand for pet food.

1.3.3 Modeling Task

Predict China’s pet food production and export trends over the next three years.

1.4 Question 4: Impact of Foreign Economic Policies

1.4.1 Objective

Analyze the impact of foreign economic policies on China’s pet industry, focusing on trade barriers and tariffs.

1.4.2 Factors

Quantitative modeling of the effects of changes in tariffs and other foreign policies on China’s pet food industry.

1.4.3 Modeling Task

Develop a model to assess the impact of these policies on China’s pet food industry and suggest strategies for sustainable development.

II、 Model Assumptions

- Stability of human society
- Exclusion of potential future policies related to pets
- Ignoring changes in pet population due to major incidents

III、 Symbol Explanation

Symbol	Meaning
ln	Logarithmic Function
$\sum$	Sum
M	Million
N	Number

IV、 Analysis

4.1 Question 1 Analysis

4.1.1 Analyze factors

The task first requires us to analyze the development trends of the pet industry in China by categorizing pets and identifying the factors that influence the growth of the pet industry. We should begin by analyzing the data.

The following data is provided to us in the problem.

Table 1 2019-2023 Number of Pet Cats and Dogs in China (in 10,000s)

Pets/Years	2023	2022	2021	2020	2019
Cat	6980	6536	5806	4862	4412
Dog	5175	5119	5429	5222	5503

Plot the trend of changes in the pet population in China. A line chart is more intuitive than a table. Thanks to the help of python, we can easily plot the line chart by the following code.

```
import pandas as pd
import matplotlib.pyplot as plt
with open('data_1.txt', 'r') as file:
   lines = file.readlines()
years = lines[0].split()[1:]
cat_data = lines[1].split()[1:]
dog_data = lines[2].split()[1:]
years = [int(year) for year in years]
cat_data = [int(num) for num in cat_data]
dog_data = [int(num) for num in dog_data]
plt.figure(figsize=(10, 6))
plt.plot(years, cat_data, marker='o', label='Cat')
```

```
plt.plot(years, dog_data, marker='o', label='Dog')
plt.xlabel('Year')
plt.ylabel('Number')
plt.title('Cat and Dog Numbers Over Years')
plt.legend()
plt.grid(True)
plt.show()
```

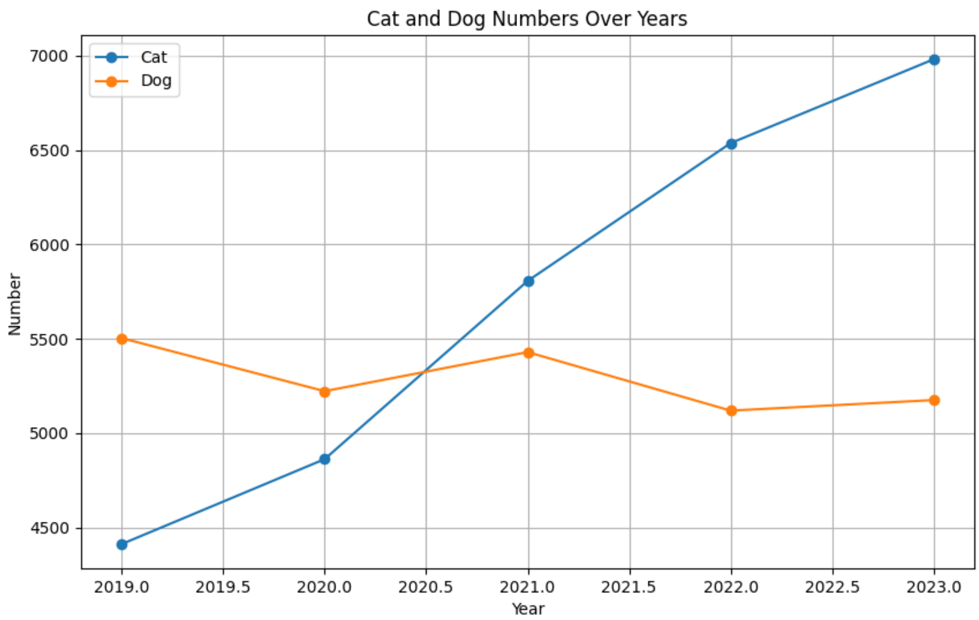

Figure 2 Number of Cat and Dog in China from 2019 to 2023

From the graph, we can see that the number of cats is increasing, while the number of dogs remains almost unchanged. This growth in the cat population is likely due to the improvement in living standards, with more and more people beginning to keep pets. Therefore, GDP is likely one of the factors influencing the pet population. By reviewing relevant data, we found a strong correlation between the growth rate of GDP and the growth rate of the pet population. Thus, to predict the growth of the pet population, we can use the growth rate of GDP as an indicator. As for dogs, we did not find a direct relationship between the GDP growth rate and the growth rate of the dog population. However, we can be certain that the development of the pet economy will drive the growth of the pet population. Hence, the pet economy is likely to have a strong correlation with the growth of the pet population.

Summarizing the above analysis, we assume that the growth of the pet population is influenced by the growth rate of GDP.

4.1.2 Polynomial Interpolation

We aim to predict the data for the next three years based on the data from the previous five years, starting with polynomial interpolation.

Definition 1 (Polynomial Interpolation) *Given $n+1$ points, polynomial interpolation refers to finding a polynomial such that it satisfies the condition*

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

That is, the image of the polynomial $y = P(x)$ must pass through the given $n+1$ points (x_i, y_i).

Because $P(x)$ can pass through these existing points and $P(x)$ is a polynomial, $P(x)$ may indeed reflect the trend of data changes to some extent.

We tried simple polynomial interpolation, with time as the independent variable and the number of cats and dogs as the dependent variable, to perform polynomial interpolation.

```
for n in range(1, 10):
    coeffs_cat = np.polyfit(years, cat_data, n)
    coeffs_dog = np.polyfit(years, dog_data, n)
    y_poly_pred_cat = np.polyval(coeffs_cat, years)
    y_poly_pred_dog = np.polyval(coeffs_dog, years)
    new_x = np.array([2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026])
    new_y_cat = np.polyval(coeffs_cat, new_x)
    new_y_dog = np.polyval(coeffs_dog, new_x)
    plt.figure(figsize=(10, 6))
    plt.plot(years, cat_data, marker='o', label='Cat')
    plt.plot(years, dog_data, marker='o', label='Dog')
```

Define k as the degree of the polynomial, n as the number of points, when $k >= n$, Function polyfit can be considered as(in fact not because of simplicity) using Lagrange interpolation to perform polynomial interpolation.

Definition 2 (Lagrange Interpolation) *Given a set of distinct nodes $x_0, x_1, \ldots, x_n$ and the corresponding function values $y_0, y_1, \ldots, y_n$, the Lagrange interpolation polynomial $L(x)$ is the polynomial that interpolates these points, defined as:*

$$L(x) = \sum_{i=0}^{n} y_i \ell_i(x)$$

where $\ell_i(x)$ is the i-th Lagrange basis polynomial, defined as:

$$\ell_i(x) = \prod_{\substack{0 \le j \le n \\ j \ne i}} \frac{x - x_j}{x_i - x_j}$$

This polynomial satisfies: $\ell_i(x_j) = \delta_{ij}$, where δ_{ij} is the Kronecker delta, which is 1 when $i = j$ and 0 otherwise.

The following is the result of the polynomial interpolation.

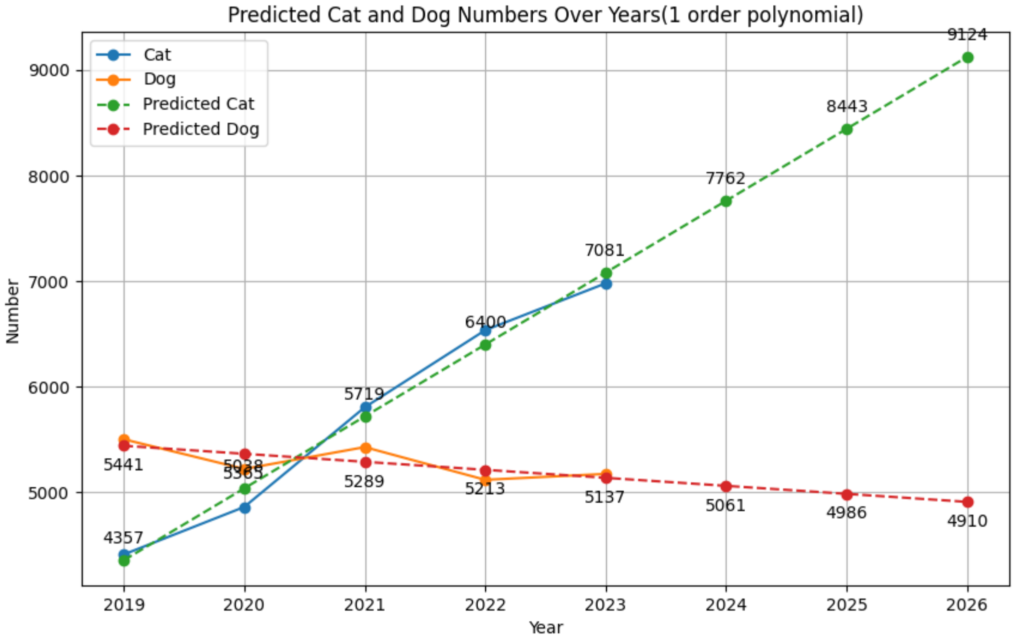

Figure 3 Linear Interpolation of Cat and Dog Numbers

We can find that although it is linear interpolation, the fitting effect is already very good. However, when the degree of the polynomial is increased, the fitting effect is not significantly improved. And because we only have five points, when the degree of the polynomial is greater than 5, the fitting effect is actually worse. To solve the above problems and apply the two influencing factors mentioned earlier, we use the following model. Its core idea is also fitting, but it is a higher dimension fitting. Its essence has not changed, but the number of independent variables has increased.

First, we collected the relevant data.

Table 2 2019-2023 GDP in China (in 10 billion yuan) [2]

Years	2023	2022	2021	2020	2019
GDP	12614	12662	12617	10408	10143

Table 3 2019-2023 Pet Industry Economy in China (in 100 million yuan) [3]

Years	2023	2022	2021	2020	2019
Pet Industry Economy	3264	3069	2733	2259	2191

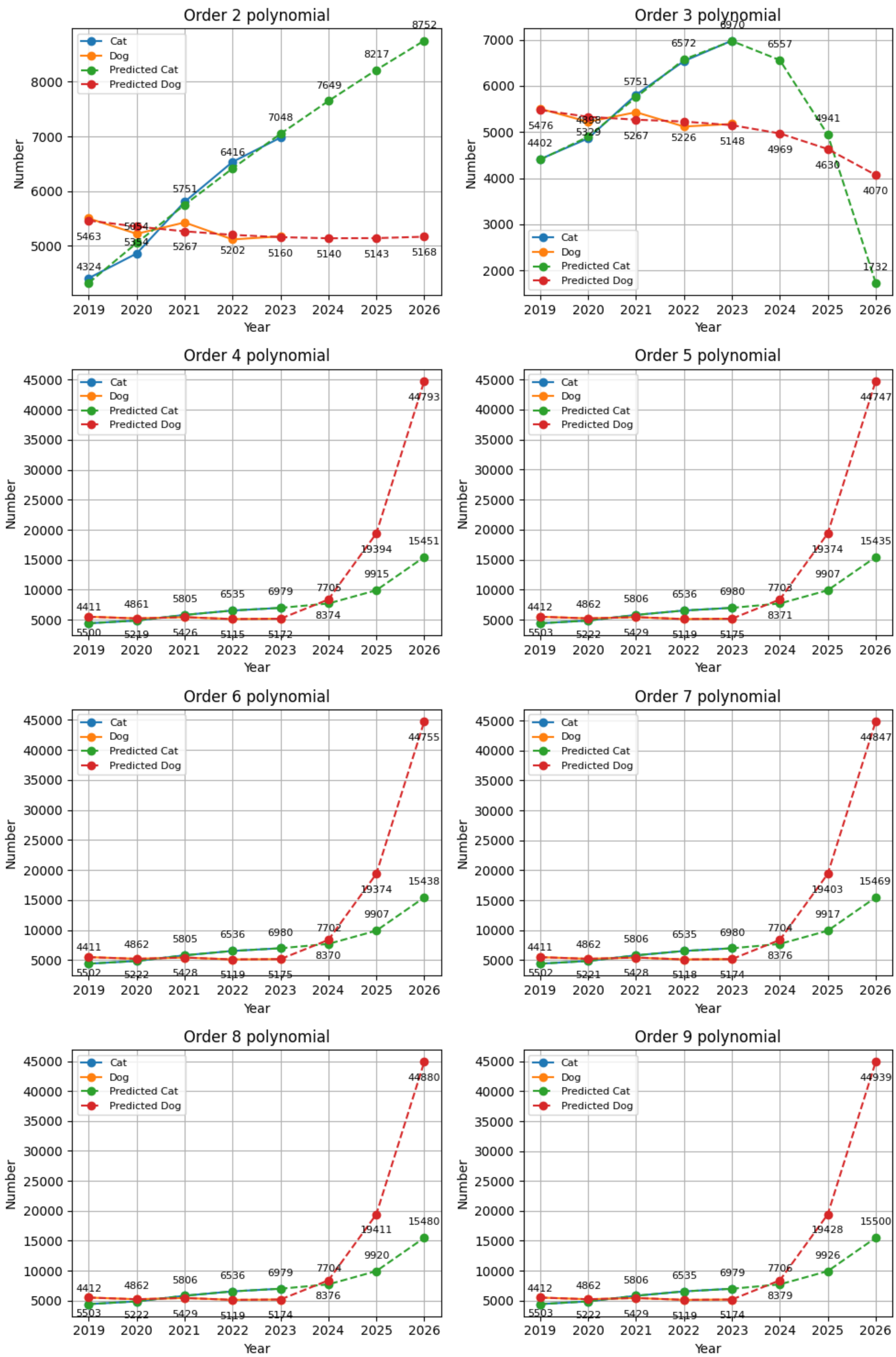

Figure 4 Polynomial Interpolation of Cat and Dog Numbers

4.1.3 Multiple Linear Regression

Similarly, in three-dimensional space, we can also find a plane that fits all the points.

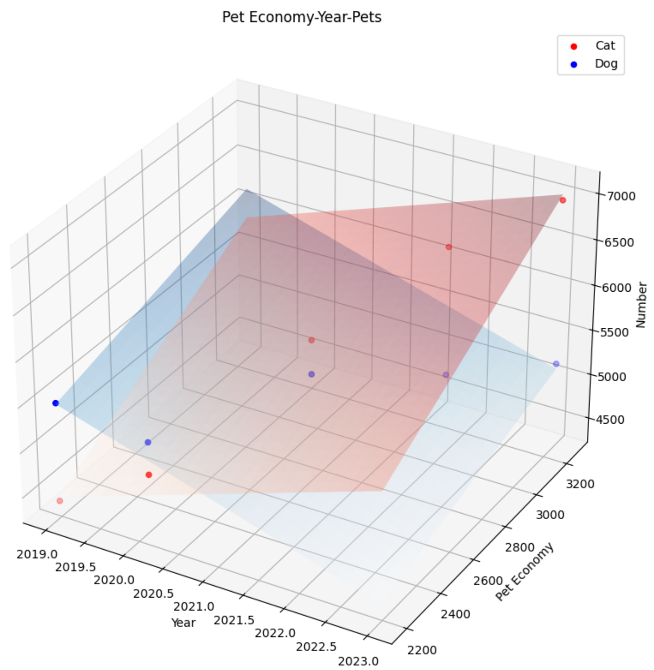

Figure 5 Three-Dimensional Polynomial Interpolation of Pet Economy-Year-Pets

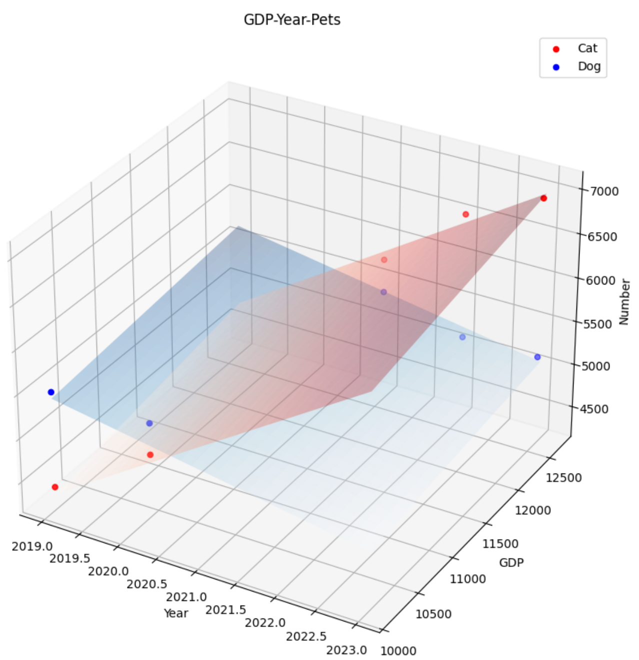

Figure 6 Three-Dimensional Polynomial Interpolation of GDP-Year-Pets

Definition 3 (Multiple Linear Regression) *Multiple Linear Regression is a statistical method used to model the relationship between two or more independent variables and a dependent vari-*

able. Suppose we have a dataset consisting of n data points, each with m independent variables and one dependent variable. The goal is to find the best-fitting hyperplane that describes the relationship between the independent and dependent variables.

Solution 1 *Assume we have n observations, each with m features. Each observation can be represented as:*

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})^T, \quad y_i$$

where y_i is the dependent variable for the i-th observation, and $\mathbf{x}_i$ is the corresponding independent variable vector. Our goal is to fit the multiple linear regression model to the following equation:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_m x_{im} + \epsilon_i$$

where β_0 is the intercept term, $\beta_1, \beta_2, \ldots, \beta_m$ are the regression coefficients, and ϵ_i is the error term, assumed to have zero mean and constant variance.

Let us represent all observations in matrix form. Suppose we have n data points, each with m features, the data can be written as:

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1m} \\ 1 & x_{21} & x_{22} & \ldots & x_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nm} \end{pmatrix}$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

where $\mathbf{X}$ is an $n \times (m+1)$ design matrix with the first column as ones, and $\mathbf{y}$ is an $n \times 1$ vector of the dependent variables.

The regression coefficients $\boldsymbol{\beta}$ are given by:

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{pmatrix}$$

Thus, the regression model in matrix form is:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon}$ is the vector of errors. To estimate the regression coefficients $\boldsymbol{\beta}$, we minimize the residual sum of squares (RSS):

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

Taking the derivative with respect to $\boldsymbol{\beta}$ and setting it equal to zero, we obtain the normal equation:

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}$$

Solving this equation gives the estimated regression coefficients:

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

```python
# key code
from sklearn.linear_model import LinearRegression
regressor_cat = LinearRegression()
regressor_cat.fit(X_cat, cat)
x_grid, y_grid = np.meshgrid(np.linspace(x.min(), x.max(), 30),
                             np.linspace(y.min(), y.max(), 30))
z_pred_cat = regressor_cat.predict(np.column_stack(
    (x_grid.ravel(), y_grid.ravel()))).reshape(x_grid.shape)
surf_cat = ax.plot_surface(
    x_grid, y_grid, z_pred_cat, alpha=0.3, cmap='Reds')
```

the problem is that we only have one independent variable, which is time, and only one dimension of data, so we can't find the corresponding points on the plane. we have to predict the data of GDP and pet industry economy so as to find the corresponding points on the plane. One good way is to use the linear regression model.

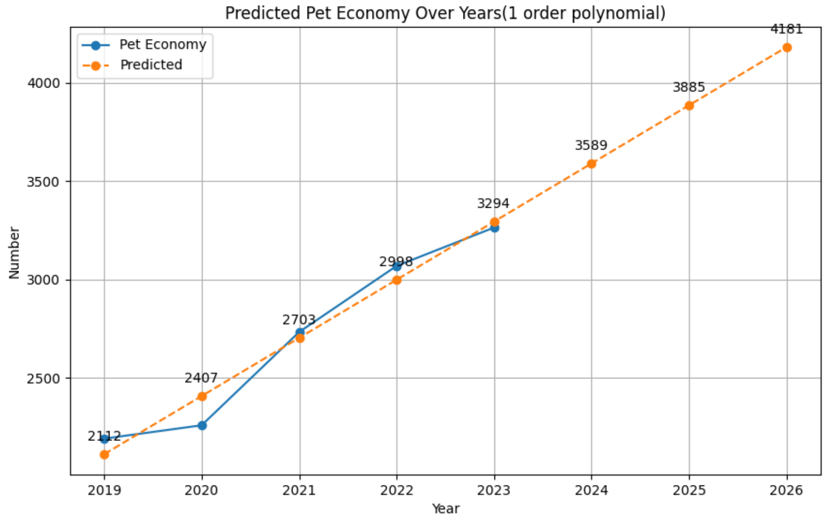

Figure 7 Linear Regression of Pet Industry Economy-Year-Pets

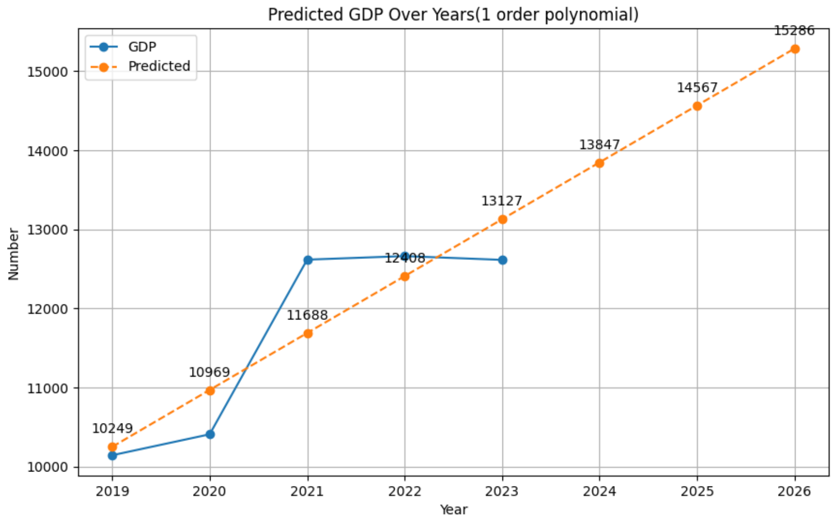

Figure 8 Linear Regression of GDP-Year-Pets

We can see that the fitting effect is very good for pet industry economy. But as for GDP, the fitting effect is a bit poor. It original data looks like a logarithm curve. Using logarithm curve to fit the data, we can get a better fitting effect.

Definition 4 (Logarithmic Curve Fitting) *The logarithmic model is expressed as:*

$$y = \sum_{i=0}^{n} a_i[\ln(bx + c)]^i = a_0 + a_1 \ln(bx + c) + a_2[\ln(bx + c)]^2 + \cdots + a_n[\ln(bx + c)]^n$$

where a_i, b, and c are constants, and x and y are the data points.

Solution 2 *To fit the logarithmic curve, we can use the following procedure. let* $z = \ln(bx + c)$*, then the model becomes:*

$$y = \sum_{i=0}^{n} a_i z^i = a_0 + a_1 z + a_2 z^2 + \cdots + a_n z^n$$

which is a polynomial model, we can use the polynomial regression model to fit the data.

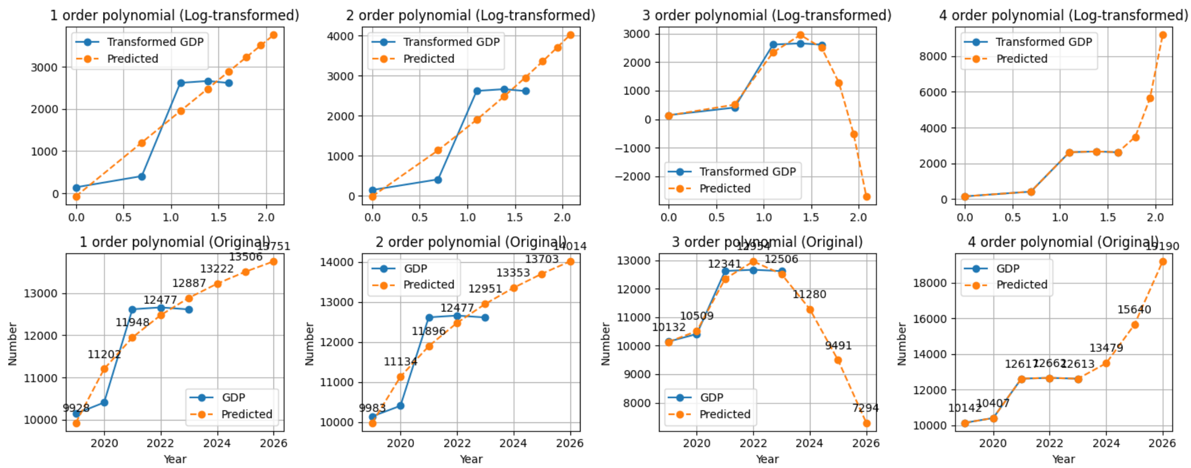

Figure 9 Logarithmic Curve Fitting of GDP

We choose the 4th order polynomial to fit the data. The reason is that China's GDP grows faster and faster despite the overfitting.

So we get predicted pet industry economy and GDP data.

Table 4 2019-2026 Predicted Pet Industry Economy in China (in 100 million yuan)

Years	2026	2025	2024	2023	2022	2021	2020	2019
Pet Industry Economy	4181	3885	3589	3294	2998	2703	2407	2112

Table 5 2019-2026 Predicted GDP in China (in 10 billion yuan)

Years	2026	2025	2024	2023	2022	2021	2020	2019
GDP	19190	15640	13479	12614	12662	12617	10408	10143

applying the two factors to the multiple linear regression model, we can get the predicted pet population.

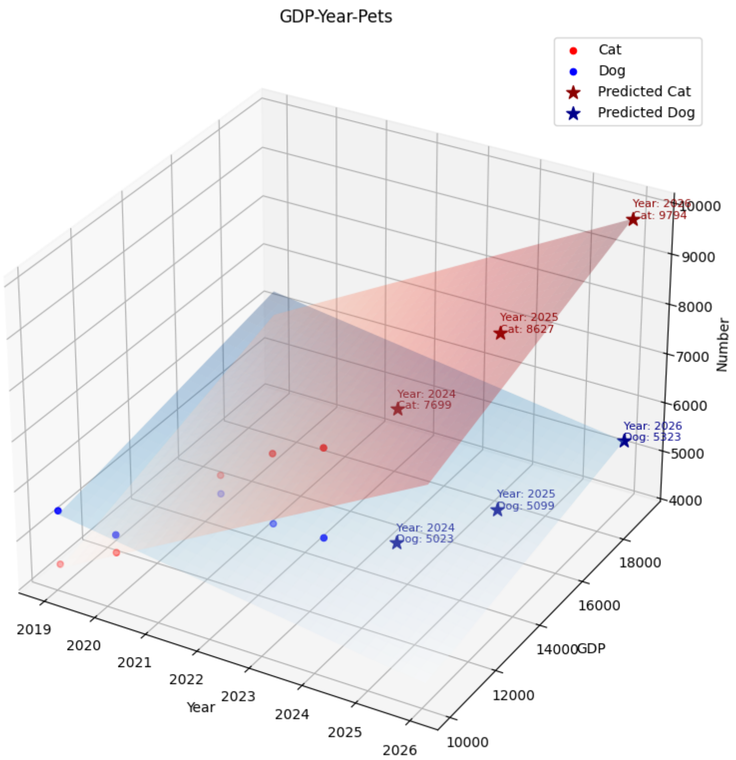

Figure 10　Predicted Pet Population in China based on GDP

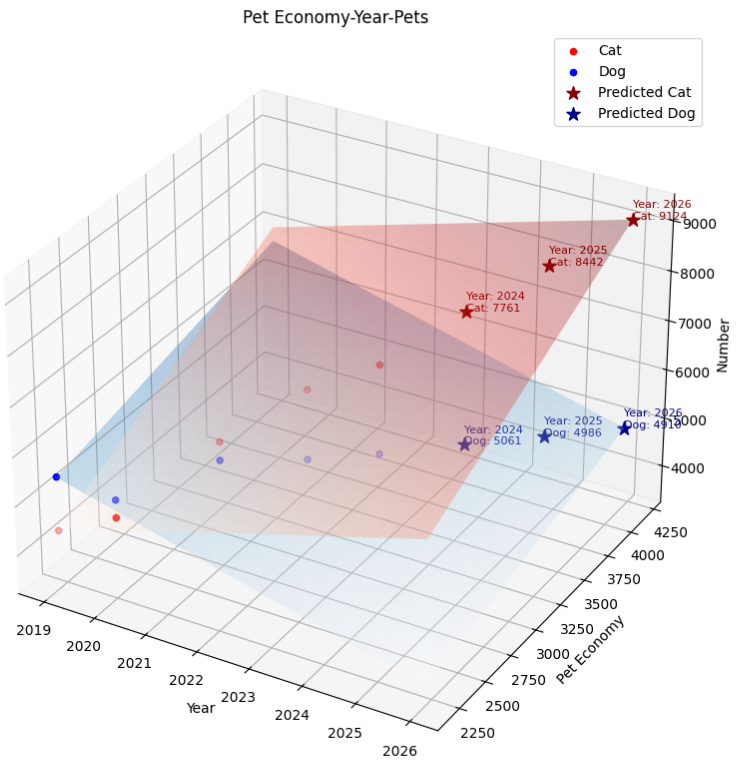

Figure 11　Predicted Pet Population in China based on Pet Industry Economy

Combine the two factors, we use 4 dimension multiple linear regression model to fit the data.

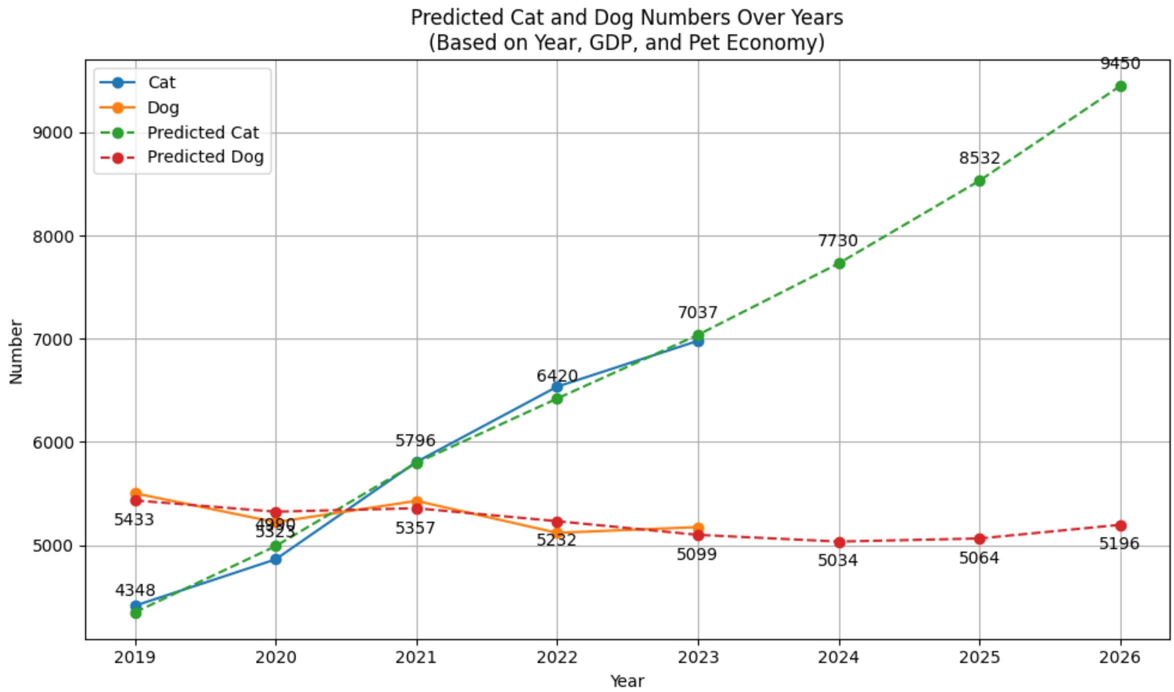

Figure 12 Predicted Pet Population in China based on GDP and Pet Industry Economy

Table 6 2019-2026 Predicted Pet Population in China (in 10000s)

Years	2026	2025	2024	2023	2022	2021	2020	2019
Cat	9450	8532	7730	6980	6536	5806	4862	4412
Dog	5196	5064	5034	5175	5119	5429	5222	5503

4.2 Question 2 Analysis

To analyze the development of the global pet industry by pet type, we first analyze the given data.

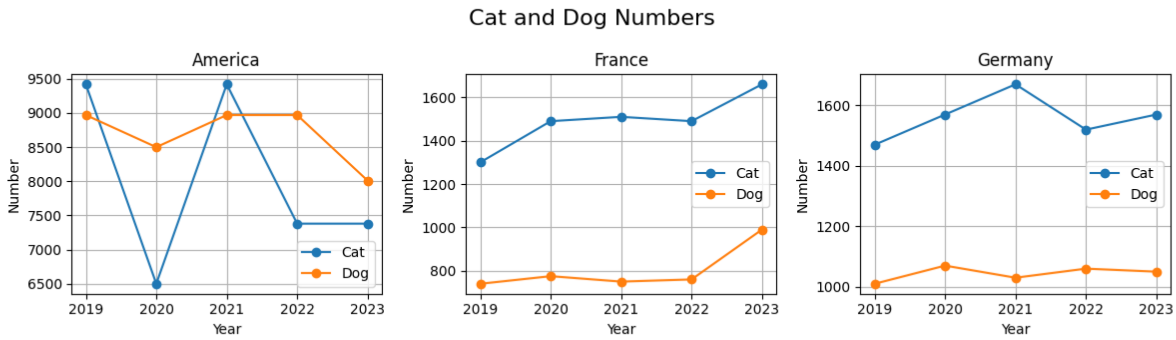

Figure 13 Cat and Dog Numbers Over Years by Country

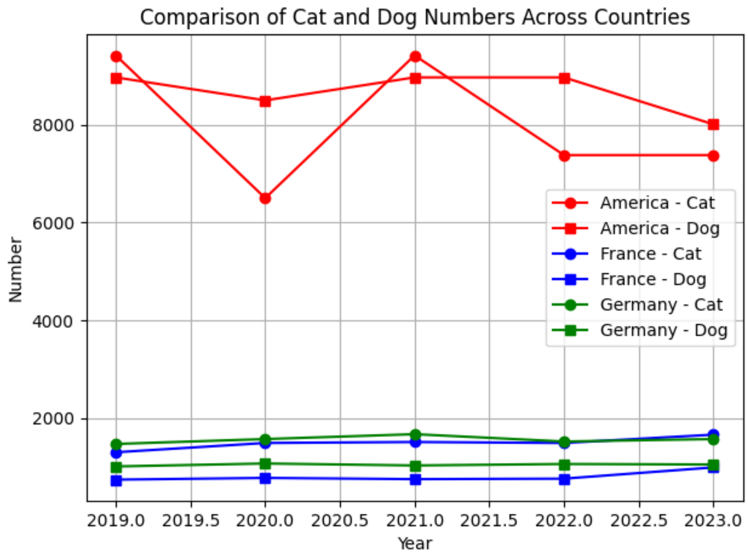

Figure 14 Cat and Dog Numbers Over Years by Country

We can see that the number of cats and dogs in America is much larger than that in France and Germany. and the number of cat and dog is almost the same.

We aim to forecast the global demand for pet food in the next three years. Because the pet food market is closely related to the pet population, we can use the predicted pet population to forecast the demand for pet food. For dataset that is time series, we can use the ARIMA model to forecast the data.

Definition 5 (ARIMA Model) *The ARIMA model, which stands for AutoRegressive Integrated Moving Average, is a popular method for forecasting time series data. It is particularly useful for predicting future points in a series based on its past values. The ARIMA model is defined by three parameters:*

$$ARIMA(p, d, q)$$

where:

- *p is the order of the autoregressive (AR) part.*
- *d is the degree of differencing required to make the time series stationary.*
- *q is the order of the moving average (MA) part.*

The AR part of the model represents a regression of the current value on its previous values. The general form for an autoregressive process of order p is:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t$$

where $\phi_1, \phi_2, \ldots, \phi_p$ are parameters, and ϵ_t is white noise.

Differencing is applied to the time series to make it stationary. If the series is not stationary, we subtract the current value from the previous value, and repeat this process d times. The differenced series is:

$$\Delta^d X_t = (1 - B)^d X_t$$

where B is the backshift operator, i.e., $B^k X_t = X_{t-k}$.

The MA part models the relationship between an observation and a residual error from a moving average model applied to lagged observations. A moving average process of order q is given by:

$$X_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}$$

where μ is the mean, ϵ_t are the white noise errors, and $\theta_1, \theta_2, \ldots, \theta_q$ are the parameters.

Combining all the components, the ARIMA model for a time series X_t is:

$$X_t = \mu + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

where μ is the mean, and the parameters ϕ_i and θ_j need to be estimated.

```python
# key code
from statsmodels.tsa.arima.model import ARIMA
cat_model = ARIMA(cat_data_array, order=(1, 1, 1))
cat_fit = cat_model.fit()
cat_pred = cat_fit.get_prediction(start=1, end=len(all_years)-1)
cat_predicted = np.concatenate(
    ([cat_data[0]], cat_pred.predicted_mean[:len(years)-1]))
cat_forecast = cat_pred.predicted_mean[len(years)-1:]
cat_all_data = np.concatenate((cat_predicted, cat_forecast))
```

Table 7 ADF and Ljung-Box Test Results

Country	Diff Order	Data Type	ADF Test	P-value	Stationarity/White Noise
America	0	Cat(ADF)	0.2371	0.7574	Non-stationary
		Dog(ADF)	0.6919	0.8653	Non-stationary
		Cat(L-B)	3.9598	0.0466	Non-white noise
		Dog(L-B)	0.7603	0.3832	White noise
America	1	Cat(ADF)	0.0000	0.6843	Non-stationary
		Dog(ADF)	0.0000	0.6843	Non-stationary
		Cat(L-B)	4.0165	0.0451	Non-white noise
		Dog(L-B)	0.1711	0.6791	White noise
America	2	Cat(ADF)	-5.3474	0.0000	Stationary
		Dog(ADF)	-1.0087	0.2846	Non-stationary
		L-B Test	Insufficient sample size		
France	0	Cat(ADF)	-1.7786	0.0716	Non-stationary
		Dog(ADF)	-1.4080	0.1483	Non-stationary
		Cat(L-B)	0.0000	1.0000	White noise
		Dog(L-B)	0.0281	0.8669	White noise
France	1	Cat(ADF)	0.0000	0.6843	Non-stationary
		Dog(ADF)	-0.0000	0.6843	Non-stationary
		Cat(L-B)	0.4705	0.4928	White noise
		Dog(L-B)	0.0167	0.8971	White noise
France	2	Cat(ADF)	-1.1357	0.2329	Non-stationary
		Dog(ADF)	-3.0520	0.0023	Stationary
		L-B Test	Insufficient sample size		
Germany	0	Cat(ADF)	-0.4659	0.5101	Non-stationary
		Dog(ADF)	-1.1406	0.2311	Non-stationary
		Cat(L-B)	0.3825	0.5362	White noise
		Dog(L-B)	3.0780	0.0794	White noise
Germany	1	Cat(ADF)	0.0000	0.6843	Non-stationary
		Dog(ADF)	-0.0000	0.6843	Non-stationary
		Cat(L-B)	0.6246	0.4294	White noise
		Dog(L-B)	3.6171	0.0572	White noise
Germany	2	Cat(ADF)	-2.4400	0.0142	Stationary
		Dog(ADF)	-18.1111	0.0000	Stationary
		L-B Test	Insufficient sample size		

We can find that most of them are not stationary, but due to the small amount of data, the confidence of the results is not high. We can still try to use the ARIMA model to predict the data. Starting with (1,1,1) for all countries, we can get the predicted data.

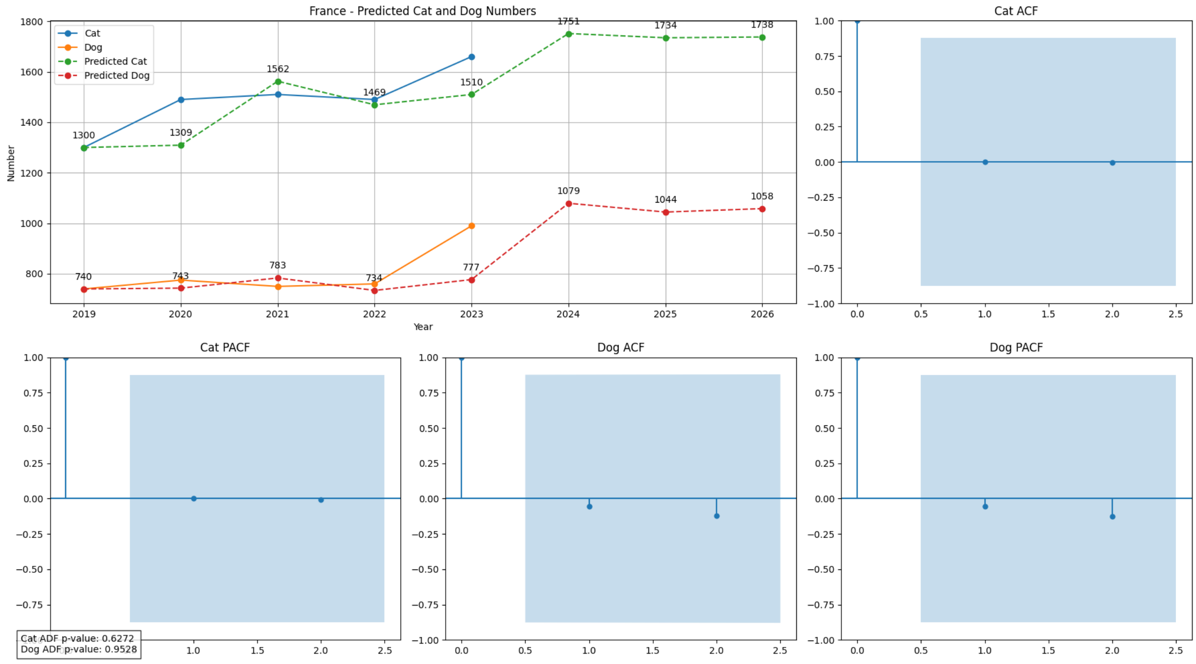

Figure 15　Predicted Cat and Dog Population in France

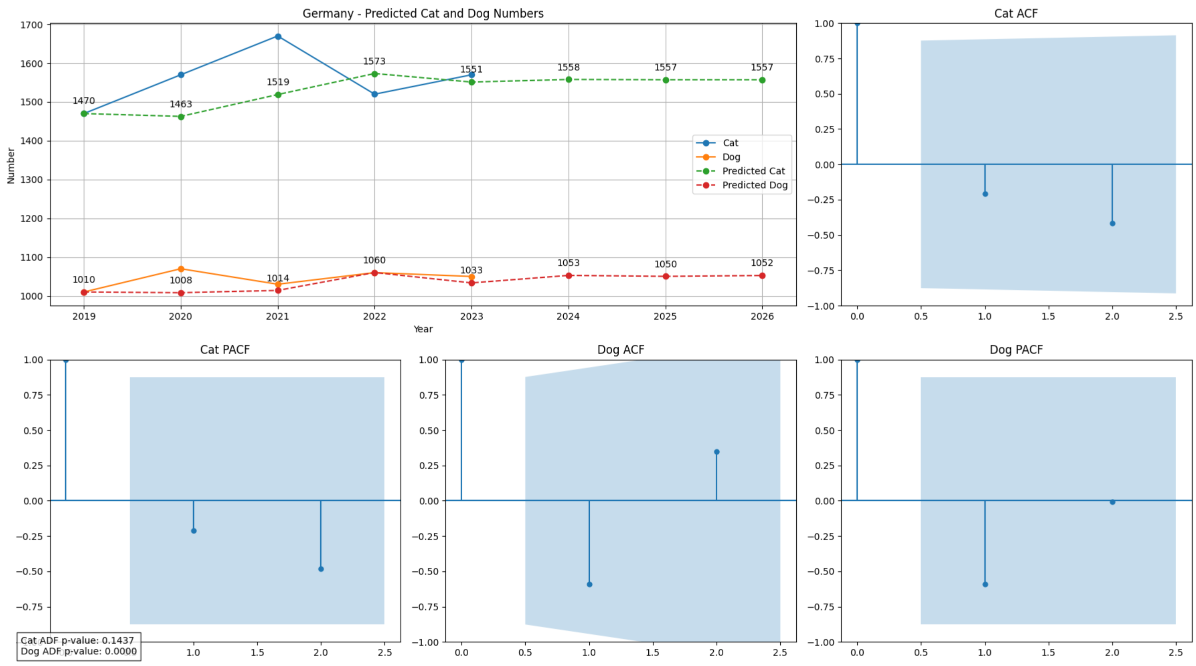

Figure 16　Predicted Cat and Dog Population in Germany

Beacause of the nearly stationary, France and Germany have a good fitting effect, we can use the predicted data to forecast the demand for pet food. However, for America, because of the non-stationary, the fitting effect is not good. So we try to use the (1,k,1) model to fit the data.

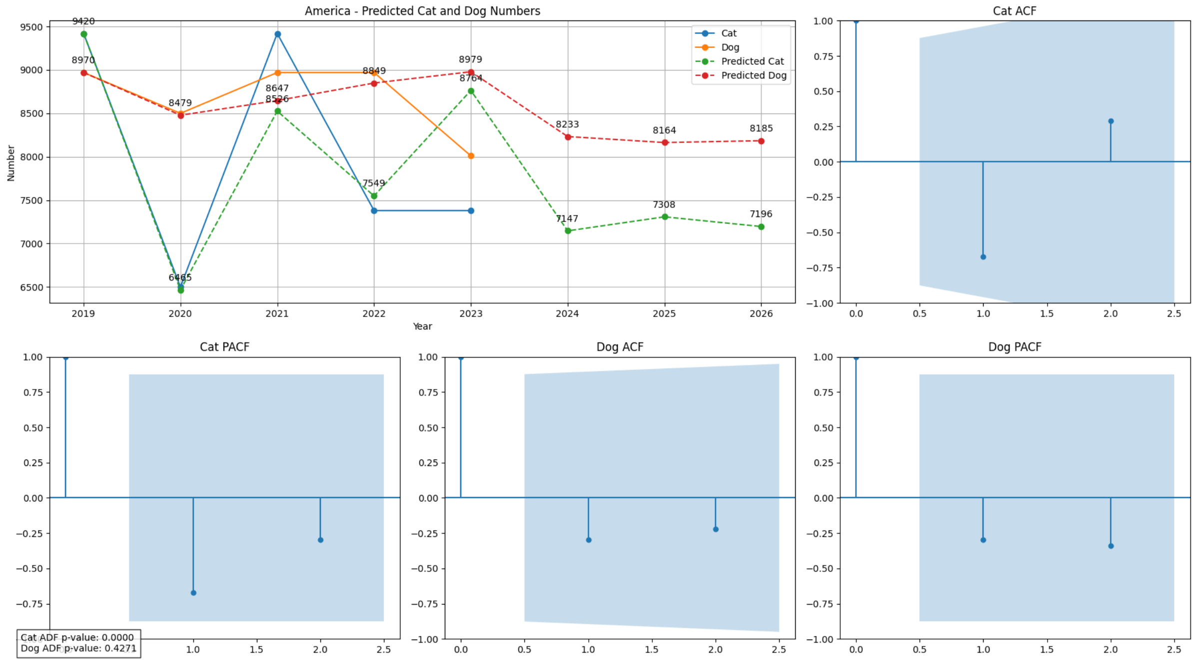

Figure 17 Predicted Cat and Dog Population in America (poor fitting effect)

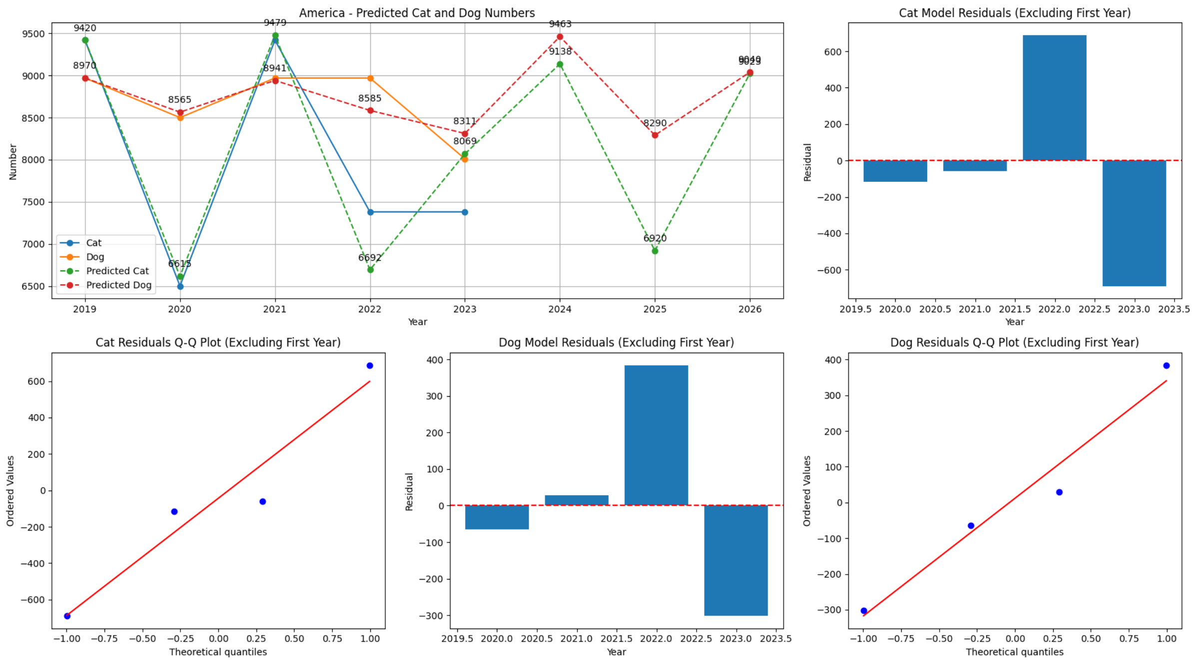

Figure 18 Predicted Cat and Dog Population in America (k=0)

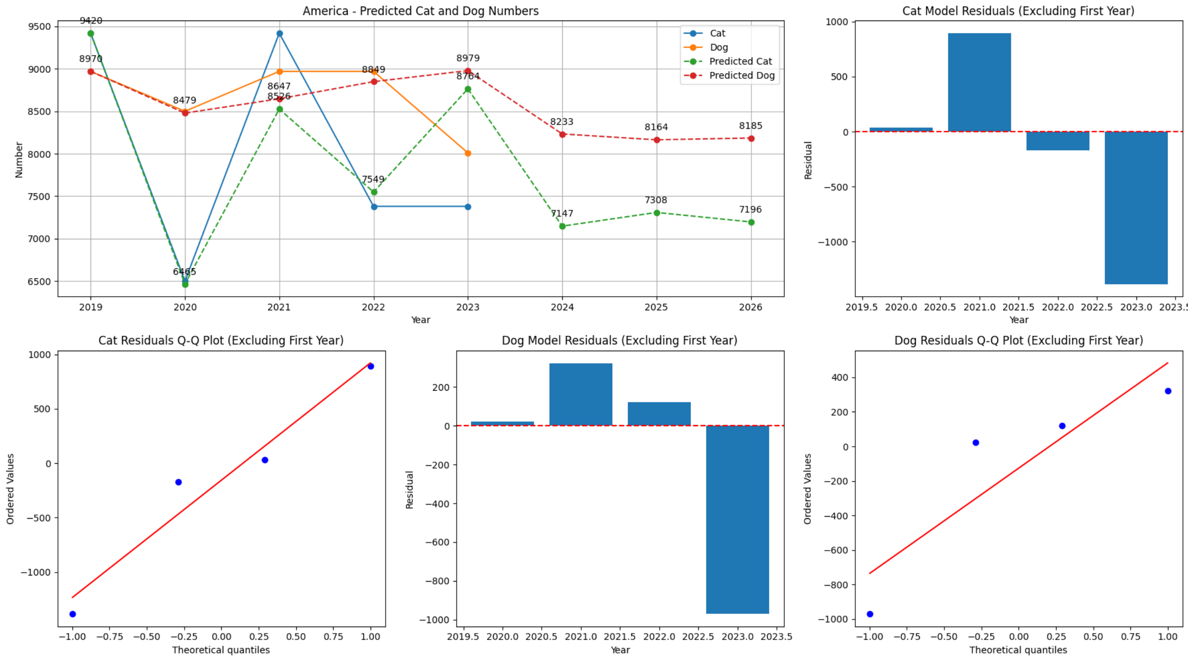

Figure 19 Predicted Cat and Dog Population in America (k=1)

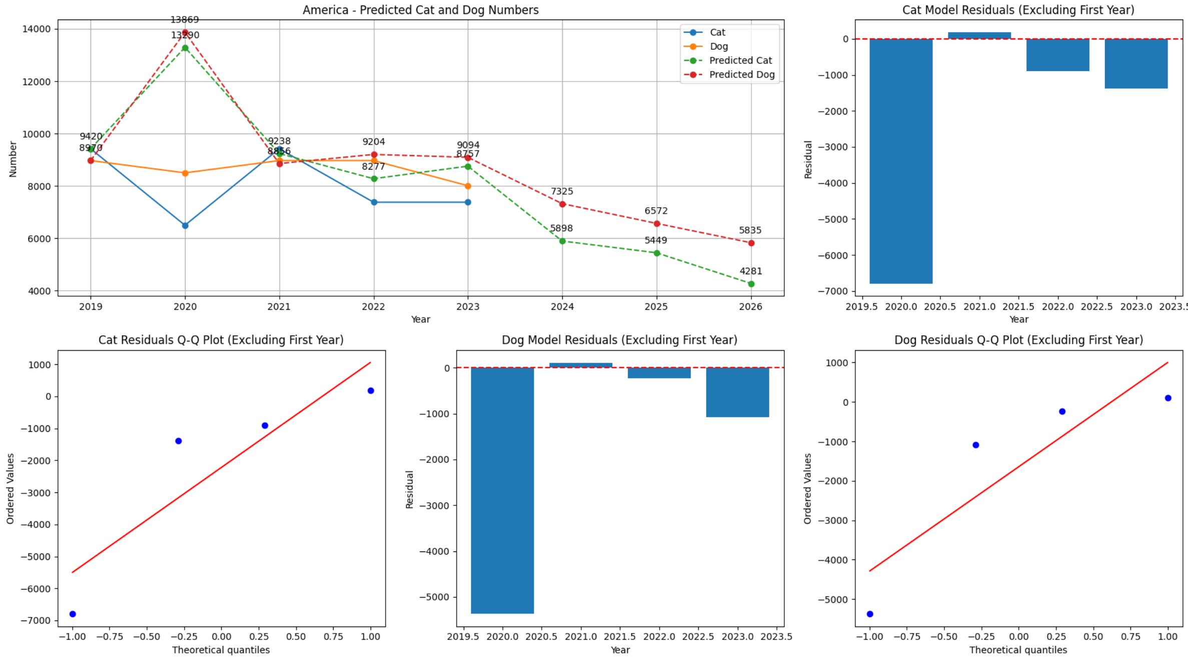

Figure 20 Predicted Cat and Dog Population in America (k=2)

We can see that when $k = 0$, the fitting effect is the best.

Table 8 MSE Evaluation Results

Differencing Order	Cat Model MSE	Dog Model MSE	Total MSE
k=0	241270.72	60995.11	302265.84
k=1	685887.84	264694.30	950582.15
k=2	12211070.97	7516437.89	19727508.86

References

[1] chendongshan. British short hair cat and golden retriever stock photo. https://www.istockphoto.com/photo/british-short-hair-cat-and-golden-retriever-gm992637094-268930508, 2024.

[2] Kylc. Global GDP Per Capita. https://www.kylc.com/stats/global/yearly_per_country/g_gdp_per_capita/chn.html, 2024.

[3] Ai Ruizixun. 2024 China Pet Industry Research Report. https://www.jiemian.com/article/11967023.html, 2024.