MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

V.N. Karazin Kharkiv National University

School of Mathematics and Computer Science

Department of theoretical and Applied informatics

Qualification work

Master

on the topic:

«Prediction of the dynamics COVID19 epidemic process

using the Logistic regression model»

Done by: 2-th year student, group MCS-64
specialty: Computer Sciences
education program: «Computer Sciences»
Wang Chao

Supervisor: Victoriya Kuznietcova
Reviewer: Kseniia Bazilevych
Adviser: Ruslan Borodai

Kharkiv, 2024

CONTENTS

ABSTRACT

The COVID-19 pandemic has brought unprecedented challenges around the world, and effective forecasting models are needed to understand its dynamics. This paper explores the use of logistic regression models to forecast COVID-19 cases in selected countries from the early stages of the pandemic to September 30, 2024. We analyze the data, perform statistical studies, and evaluate the performance of the model over different forecast periods. The results reveal the accuracy of the model and its impact on public health strategies[1][4].The results of this study have important implications for public health strategies, especially when responding to infectious diseases such as COVID-19. The predictive insights provided by logistic regression models can enable policymakers and public health officials to implement precision interventions more efficiently. First, predictive models can help identify trends in the early stages of an epidemic, allowing governments to stockpile medical supplies and prepare medical facilities in advance, thereby improving response efficiency. Second, accurate predictions can optimize resource allocation, ensure that high-risk areas receive adequate medical support, and reduce the pressure on the medical system. In addition, the prediction results can guide the implementation of precision interventions such as local blockades, travel restrictions, or vaccinations to reduce the widespread impact on the economy and society. At the same time, the model can also optimize vaccine distribution strategies to ensure that limited vaccine resources are used efficiently by predicting future epidemic hotspots and high-risk

populations. Transparent prediction results can help enhance public trust and cooperation in health measures. For example, by showing the effects of social distancing or wearing masks through data, the public's willingness to comply can be increased. Finally, predictive models can also support policy evaluation and adjustment, helping public health departments to analyze the effectiveness of strategies in real time and optimize them in a timely manner. At the same time, they also provide a scientific basis for long-term planning for future epidemics and enhance overall resilience and response capabilities. In these ways, predictive models provide solid scientific support for the development and implementation of public health strategies.

1. INTRODUCTION

1.1 Background Introduction

Since the outbreak of COVID-19 at the end of 2019, public health and economic activities around the world have been unprecedentedly affected. Governments and health organizations are under tremendous pressure to make quick decisions to curb the spread of the virus. As the epidemic develops, it becomes particularly important to accurately predict the dynamic process of the epidemic, which not only helps to formulate effective prevention and control measures but also provides a scientific basis for the allocation of medical resources[2][3][4].

1.2 Importance of the study

In the process of responding to the epidemic, although traditional epidemiological models provide a basic prediction framework, a single model often fails to meet the needs when data complexity and variability are increasing. Logistic regression model, as a statistical method widely used in binary classification problems, can effectively handle nonlinear relationships and provide interpretable results. Therefore, using logistic regression model to dynamically predict the COVID-19 epidemic can not only improve the accuracy of prediction, but also provide empirical support for public health decision-making.

1.3 Research Objectives and Problem Statement

The study aims to use the logistic regression model to predict the dynamic process of the COVID-19 epidemic. Specific objectives include:

1. Collect and analyze relevant epidemic data to build an effective prediction model;

2. Evaluate the effectiveness and reliability of the logistic regression model in predicting the COVID-19 epidemic;

3. Explore the impact of the model prediction results on public health policies and their application value.

2. LITERATURE REVIEW

2.1 Overview of COVID-19 related research

Since the outbreak of COVID-19, researchers around the world have rapidly conducted a large number of studies to understand the mechanisms and influencing factors of virus transmission. Many studies focus on the transmission characteristics, infection rates, and mortality rates of the epidemic. For example, some studies use time series analysis and epidemiological models to explore the spread of the epidemic in different countries and regions. In addition, the application of social media and mobile data has also been included in the research to reveal the impact of population mobility on the development of the epidemic. These studies provide important references for the formulation of public health policies, but most models are still insufficient when faced with complex epidemic dynamics.

2.2 Application history of logistic regression model

The logistic regression model was first proposed by Bernoulli in the 19th century. As a statistical method for binary classification problems, logistic regression has been widely used in social sciences, medicine, economics and other fields. In the field of public health, logistic regression is often used to analyze risk factors affecting the occurrence of diseases and to predict diseases. In recent years, with the development of big data technology, logistic regression models have also been gradually applied to epidemiological research, especially in the prediction of COVID-19 epidemics. Studies have shown that logistic regression models can effectively capture nonlinear relationships in epidemic data and provide intuitive risk assessment results.

2.3 Comparison of other prediction models

In the prediction of the COVID-19 epidemic, in addition to the logistic regression model, researchers have also adopted a variety of other models, such as the SIR (susceptible-infected-recovered) model, the ARIMA (autoregressive integrated moving average) model, and machine learning algorithms (such as random forests and neural networks). The SIR model describes the spread of infectious diseases through mathematical equations, which is suitable for long-term predictions, but may not perform well in short-term dynamic changes. The ARIMA model focuses more on the trend and seasonality of time series data, but its ability to handle external variables is limited. In contrast, although machine learning algorithms perform well in processing complex data, they usually lack interpretability[5].

Among these models, the advantage of logistic regression lies in its simplicity and interpretability, which enables researchers to clearly understand the key factors affecting the spread of the epidemic. Therefore, it is of great theoretical and practical significance to combine the advantages of logistic regression and other models and explore their application in the dynamic prediction of the COVID-19 epidemic.

3. METHODOLOGY

3.1 Data Collection

3.1.1 Data Source

The data for this study mainly comes from the Global Epidemic Database (Our World in Data), specifically focusing on the COVID-19 epidemic data in China. The dataset contains several important fields, among which the "total_cases" field records the cumulative number of infected cases per day[1][6].

3.1.2 Data features

In the dataset, the main features analyzed include:

Total number of infections (total_cases): indicates the cumulative number of confirmed cases since the beginning of the epidemic.

New number of infections (new_cases): the number of newly confirmed cases per day.

Total number of deaths (total_deaths): indicates the cumulative number of deaths since the beginning of the epidemic.

New number of deaths (new_deaths): the number of newly added deaths per day.

These features provide the necessary inputs for the logistic regression model to predict the dynamics of the epidemic.

3.2 Data preprocessing

Before data analysis, the raw data was cleaned and processed.

3.2.1 Data cleaning and processing methods

Remove missing values: For missing "total_cases" or "date" fields, delete the corresponding records.

Format conversion: Ensure that the date format is in the standard YYYY/MM/DD format to facilitate subsequent analysis.

3.2.2 Feature selection

“total_cases” and “new_cases” were selected as the main analysis features, and other features such as “total_deaths” and “new_deaths” could be used for subsequent model evaluation.

3.3 Logistic regression model construction

The logistic regression model is a statistical model for binary classification problems, suitable for analyzing the relationship between independent variables and dependent variables. In this study, the model will be used to predict the number of future infections[4][5].

3.3.1 Theoretical basis of the model

The logistic regression model is modeled by log-odds, and and the formula is :

$$logit(p) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n \tag{1.1}$$

Among them, p is the probability of an event occurring, Xi is the independent variable, and βi is the model parameter.

3.3.2 Model formula and parameter setting

In this study, "total_cases" was selected as the dependent variable, and new_cases, new_cases_smoothed, total_deaths, new_deaths, new_deaths_smoothed were used as independent variables. The model parameters were estimated by maximum likelihood estimation (MLE).

3.4 Model training and validation

3.4.1 Division of training set and test set

The dataset is divided into the training set and test set:

Based on data from the start of the epidemic to July 30, 2024:

Training set: data from the start of the epidemic to a specific date (for example, when predicting 3 days, the training set can be data from the start of the epidemic to July 27, 2024).

Test set: data from a few days immediately following the training set (for example, when predicting 3 days, the test set is data from July 28 to 30, 2024).

Based on data from January 1, 2024 to July 30, 2024:

Training set: data from January 1, 2024 to a specific date (for example, when predicting 3 days, the training set can be data from the start of the epidemic to July 27, 2024).

Test set: data from a few days immediately following the training set (for example, when predicting 3 days, the test set is data from July 28 to 30, 2024).

3.4.2 Model evaluation metrics

Absolute Error: This is the absolute difference between the predicted value and the actual value. It is calculated as:

$$Absolute\ Error = |\ Actual\ Value - Predicted\ Value\ |$$

Relative error: This refers to the ratio of the absolute error to the actual value, usually expressed as a percentage. The calculation method is:

$Relative\ \ Error\ =\ \ |\ Actual\ \ Value\ -\ Predicted\ \ Value\ |\ \ \div\ Actual\ \ Value\ \times\ 10$

0%

4. RESULTS

4.1 Model prediction results of the whole process

4.1.1 Prediction charts and data analysis of the entire process

Conduct a statistical study (to train the model, use data from the first day of the pandemic in the selected country) according to the scheme:

Use a logistic regression model, select "total_cases" as the dependent variable and use new_cases, new_cases_smoothed, total_deaths, new_deaths, new_deaths_smoothed as the independent variables.

Use all the data from the start date to the end date as training data, output the predicted results, draw a graph, compare them with the actual data, and calculate the absolute error and relative error.

Table 4.1 − Prediction Periods and Absolute Error Calculation:

	Actual data end date	Dates included in prediction	Calculate the absolute prediction error
Prediction for 3 days	27.07.24	2024 July 28-30	Absolute error: real data as of September 30 minus data as of September 30 produced by the model, modulo $\Delta a \geq \lvert A - a \rvert$, so $a - \Delta a \leq A \leq a + \Delta a$ or $A = a \pm \Delta a.$

Continuation Table 4.1 – Prediction Periods and Absolute Error Calculation:

Prediction for 5 days	25.07.24	2024 July 26-30	
Prediction for 7 days	23.07.24	2024 July 24-30	
Prediction for 10 days	20.07.24	2024 July 21-30	
Prediction for 14 days	16.07.24	2024 July 17-30	
Prediction for 21 days	09.07.24	2024 July 10-30	
Prediction for 30 days	31.07.24	July 1-30	

Table 4.2- Prediction Periods and Relative Error Calculation :

<table>
<tr><th></th><th>Actual data end date</th><th>Dates included in prediction</th><th>Calculate the relative prediction error</th></tr>
<tr><td>Prediction for 3 days</td><td>27.07.24</td><td>2024 July 28-30</td><td rowspan="5">Relative error:
$\delta a = \frac{|A - a|}{|a|}$ or $\delta a = \frac{\Delta a}{|a|}$.</td></tr>
<tr><td>Prediction for 5 days</td><td>25.07.24</td><td>2024 July 26-30</td></tr>
<tr><td>Prediction for 7 days</td><td>23.07.24</td><td>2024 July 24-30</td></tr>
<tr><td>Prediction for 10 days</td><td>20.07.24</td><td>2024 July 21-30</td></tr>
<tr><td>Prediction for 14 days</td><td>16.07.24</td><td>2024 July 17-30</td></tr>
</table>

Continuation Table 4.2- Prediction Periods and Relative Error Calculation :

Prediction for 21 days	09.07.24	2024 July 10-30	
Prediction for 30 days	31.07.24	July 1-30	

Here is the Python code：

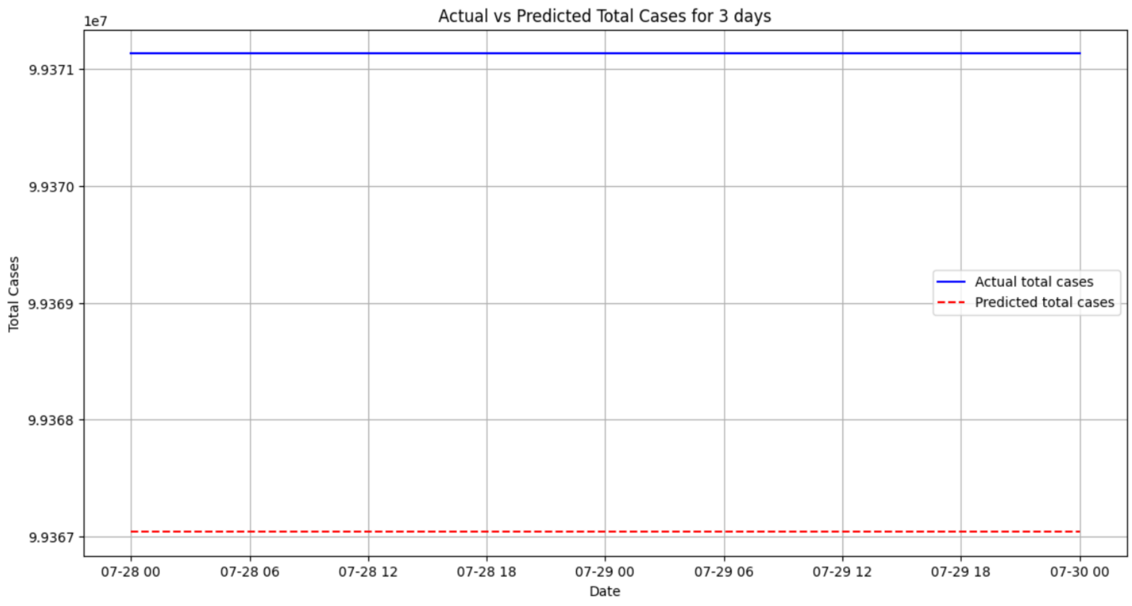
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Load data
data = pd.read_csv('China_total_amount.csv')

# Convert date format
data['date'] = pd.to_datetime(data['date'])

# Fill missing values
data.fillna(method='ffill', inplace=True)

# Select features and target variable
features = ['new_cases', 'new_cases_smoothed', 'total_deaths', 'new_deaths',
'new_deaths_smoothed']
X = data[features]
y = data['total_cases']
```

```
# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train logistic regression model
model = LogisticRegression(max_iter=10000)
model.fit(X_scaled, y)

# Define prediction date ranges
prediction_ranges = {
    "3 days": ('2024-07-28', '2024-07-30'),
    "5 days": ('2024-07-26', '2024-07-30'),
    "7 days": ('2024-07-24', '2024-07-30'),
    "10 days": ('2024-07-21', '2024-07-30'),
    "14 days": ('2024-07-17', '2024-07-30'),
    "21 days": ('2024-07-10', '2024-07-30'),
    "30 days": ('2024-07-01', '2024-07-30')
}

for key, (start_date, end_date) in prediction_ranges.items():
    # Select specific date range for prediction
    date_mask = (data['date'] >= start_date) & (data['date'] <= end_date)
    prediction_data = data[date_mask]
    X_pred = prediction_data[features]
    X_pred_scaled = scaler.transform(X_pred)
    predictions = model.predict(X_pred_scaled)

    # Calculate errors
    actual = prediction_data['total_cases'].values
    absolute_error = np.abs(predictions - actual)
```

```
    relative_error = np.abs(predictions - actual) / actual


    # Output prediction results and errors
    prediction_data['Predicted_total_cases'] = predictions
    prediction_data['Absolute_Error'] = absolute_error
    prediction_data['Relative_Error'] = relative_error
    print(f"\nPrediction for {key}:")
    print(prediction_data[['date', 'total_cases', 'Predicted_total_cases', 'Absolute_Error',
'Relative_Error']])


    # Plot comparison of predicted results and actual data
    plt.figure(figsize=(14, 7))
    plt.plot(prediction_data['date'], prediction_data['total_cases'], label='Actual total
cases', color='blue')
    plt.plot(prediction_data['date'], prediction_data['Predicted_total_cases'],
label='Predicted total cases', color='red', linestyle='--')
    plt.xlabel('Date')
    plt.ylabel('Total Cases')
    plt.title(f'Actual vs Predicted Total Cases for {key}')
    plt.legend()
    plt.grid(True)
    plt.show()
```

Table 4.3 - Prediction for 3 days:

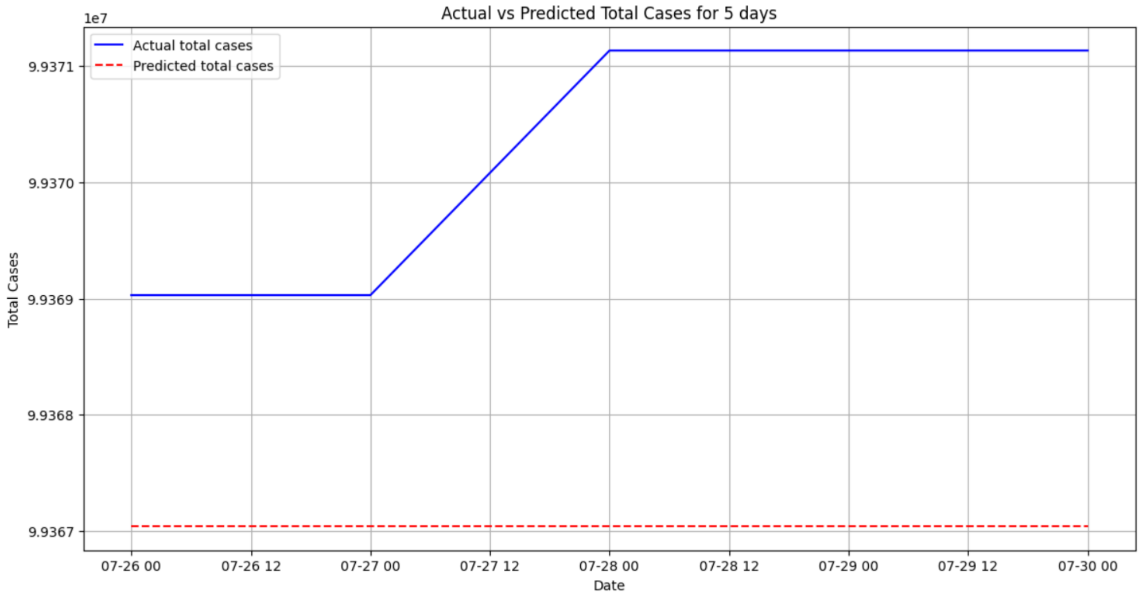
Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-28	99371132	99367041	4091	0.000041
2024-07-29	99371132	99367041	4091	0.000041
2024-07-30	99371132	99367041	4091	0.000041



Graph 4.1 - Actual vs Predicted Total Cases for 3 days

Table 4.4 - Prediction for 5 days:

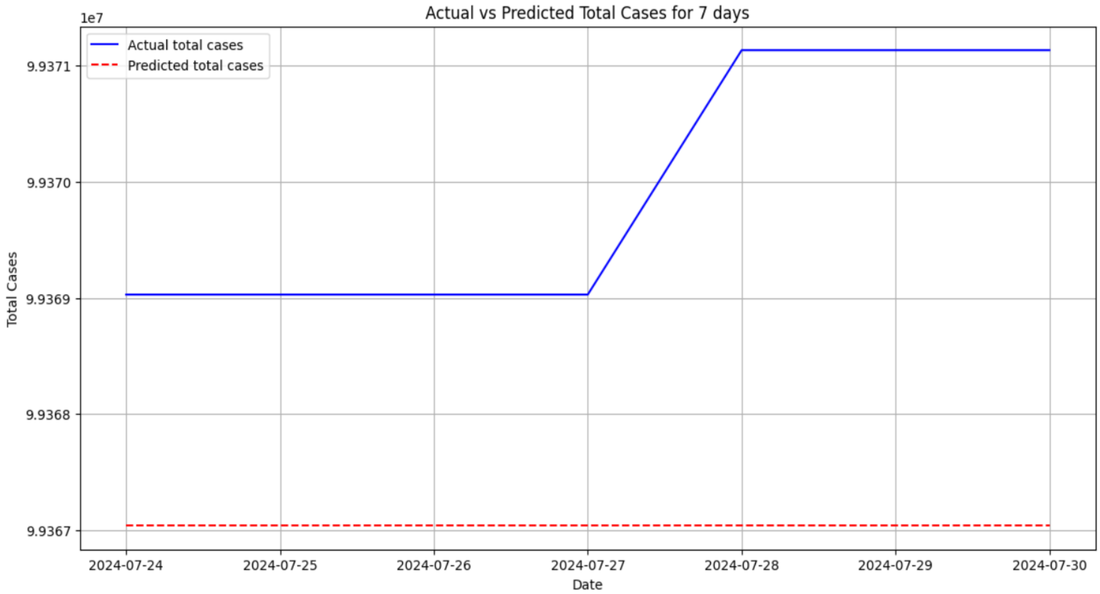
Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-26	99369029	99367041	1988	0.000020
2024-07-27	99369029	99367041	1988	0.000020
2024-07-28	99371132	99367041	4091	0.000041
2024-07-29	99371132	99367041	4091	0.000041
2024-07-30	99371132	99367041	4091	0.000041



Graph 4.2 - Actual vs Predicted Total Cases for 5 days

Table 4.5 - Prediction for 7 days:

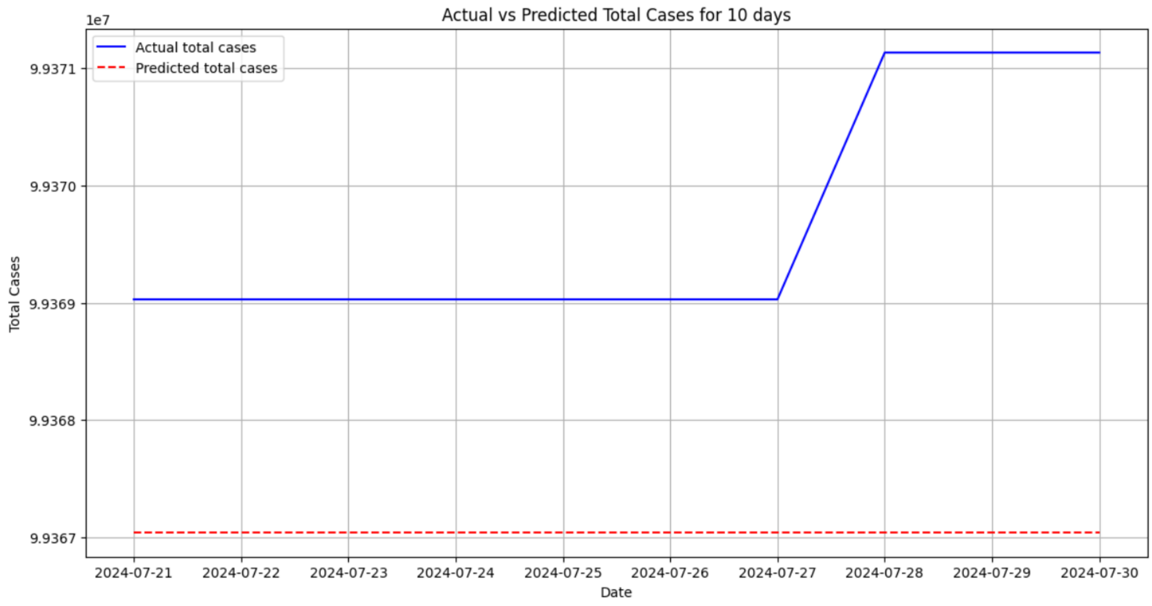
Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-24	99369029	99367041	1988	0.000020
2024-07-25	99369029	99367041	1988	0.000020
2024-07-26	99369029	99367041	1988	0.000020
2024-07-27	99369029	99367041	1988	0.000020
2024-07-28	99371132	99367041	4091	0.000041
2024-07-29	99371132	99367041	4091	0.000041
2024-07-30	99371132	99367041	4091	0.000041



Graph 4.3 - Actual vs Predicted Total Cases for 7 days

Table 4.6 - Prediction for 10 days:

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-21	99369029	99367041	1988	0.000020
2024-07-22	99369029	99367041	1988	0.000020
2024-07-23	99369029	99367041	1988	0.000020
2024-07-24	99369029	99367041	1988	0.000020
2024-07-25	99369029	99367041	1988	0.000020
2024-07-26	99369029	99367041	1988	0.000020
2024-07-27	99369029	99367041	1988	0.000020
2024-07-28	99371132	99367041	4091	0.000041
2024-07-29	99371132	99367041	4091	0.000041
2024-07-30	99371132	99367041	4091	0.000041



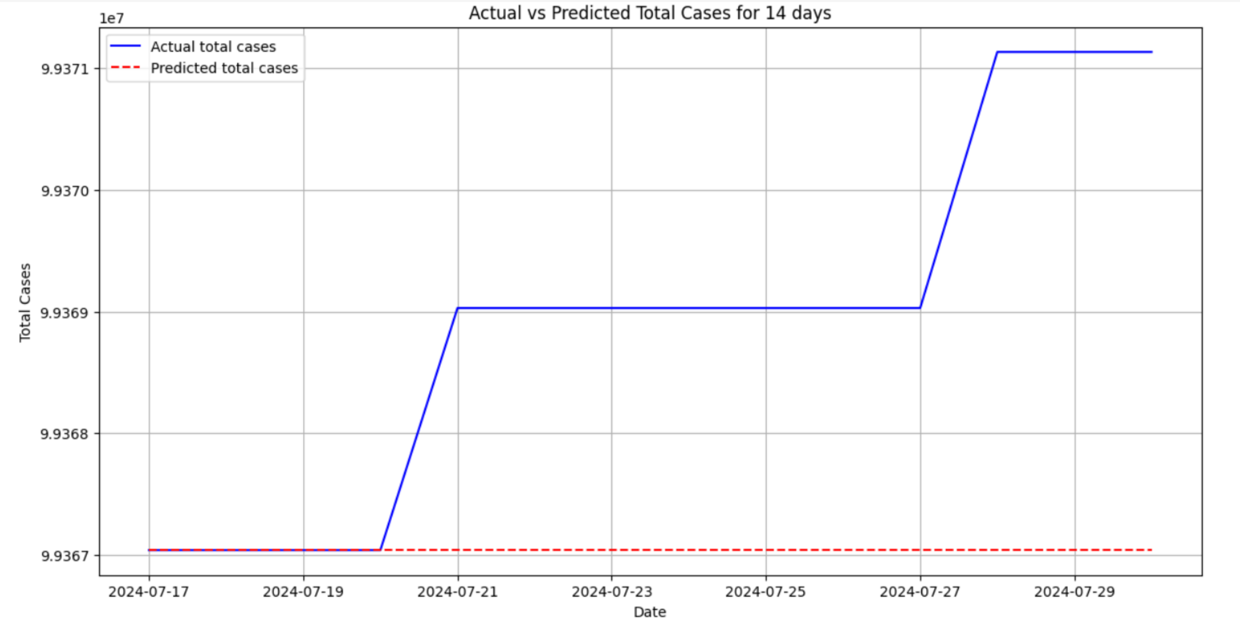
Graph 4.4 - Actual vs Predicted Total Cases for 10 days

Table 4.7 - Prediction for 14 days:

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-17	99367041	99367041	0	0.000000
2024-07-18	99367041	99367041	0	0.000000
2024-07-19	99367041	99367041	0	0.000000
2024-07-20	99367041	99367041	0	0.000000
2024-07-21	99369029	99367041	1988	0.000020
2024-07-22	99369029	99367041	1988	0.000020
2024-07-23	99369029	99367041	1988	0.000020
2024-07-24	99369029	99367041	1988	0.000020
2024-07-25	99369029	99367041	1988	0.000020

Continuation Table 4.7 - Prediction for 14 days:

2024-07-26	99369029	99367041	1988	0.000020
2024-07-27	99369029	99367041	1988	0.000020
2024-07-28	99371132	99367041	4091	0.000041
2024-07-29	99371132	99367041	4091	0.000041
2024-07-30	99371132	99367041	4091	0.000041



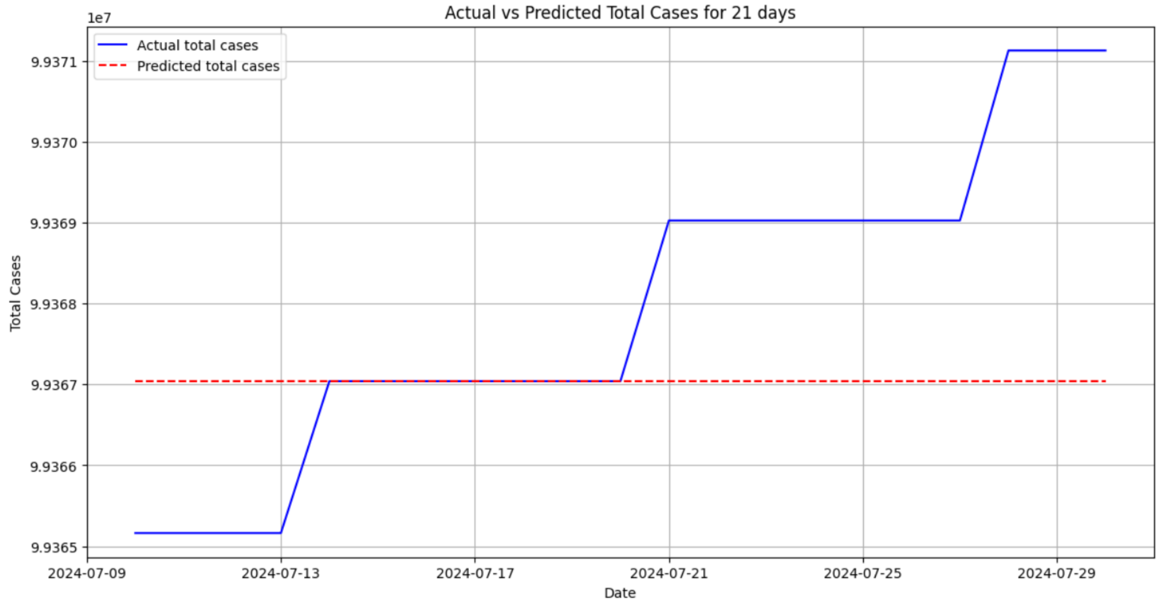
Graph 4.5 - Actual vs Predicted Total Cases for 14 days

Table 4.8 - Actual vs Predicted Total Cases for 21 days

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-10	99365162	99367041	1879	0.000019
2024-07-11	99365162	99367041	1879	0.000019

Continuation Table 4.8 - Actual vs Predicted Total Cases for 21 days

2024-07-12	99365162	99367041	1879	0.000019
2024-07-13	99365162	99367041	1879	0.000019
2024-07-14	99367041	99367041	0	0.000000
2024-07-15	99367041	99367041	0	0.000000
2024-07-16	99367041	99367041	0	0.000000
2024-07-17	99367041	99367041	0	0.000000
2024-07-18	99367041	99367041	0	0.000000
2024-07-19	99367041	99367041	0	0.000000
2024-07-20	99367041	99367041	0	0.000000
2024-07-21	99369029	99367041	1988	0.000020
2024-07-22	99369029	99367041	1988	0.000020
2024-07-23	99369029	99367041	1988	0.000020
2024-07-24	99369029	99367041	1988	0.000020
2024-07-25	99369029	99367041	1988	0.000020
2024-07-26	99369029	99367041	1988	0.000020
2024-07-27	99369029	99367041	1988	0.000020



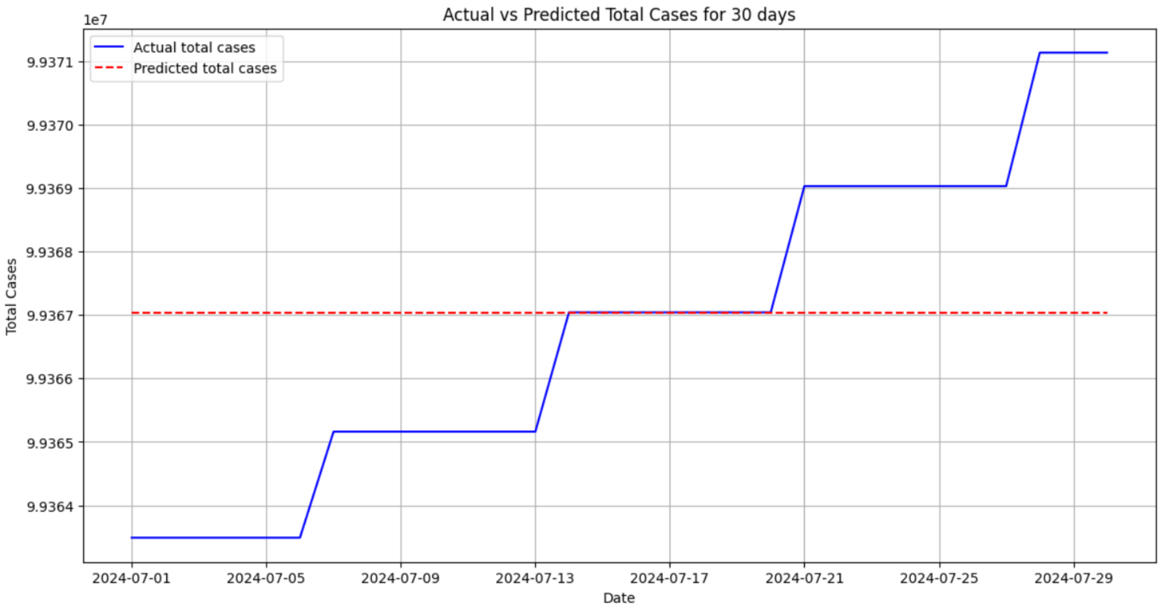
Graph 4.6 - Actual vs Predicted Total Cases for 21 days

Table 4.9 - Prediction for 30 days:

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-01	99363491	99367041	3550	0.000036
2024-07-02	99363491	99367041	3550	0.000036
2024-07-03	99363491	99367041	3550	0.000036
2024-07-04	99363491	99367041	3550	0.000036
2024-07-05	99363491	99367041	3550	0.000036
2024-07-06	99363491	99367041	3550	0.000036
2024-07-07	99365162	99367041	1879	0.000019
2024-07-08	99365162	99367041	1879	0.000019
2024-07-09	99365162	99367041	1879	0.000019
2024-07-10	99365162	99367041	1879	0.000019

Continuation Table 4.9 - Prediction for 30 days:

2024-07-11	99365162	99367041	1879	0.000019
2024-07-12	99365162	99367041	1879	0.000019
2024-07-13	99365162	99367041	1879	0.000019
2024-07-14	99367041	99367041	0	0.000000
2024-07-15	99367041	99367041	0	0.000000
2024-07-16	99367041	99367041	0	0.000000
2024-07-17	99367041	99367041	0	0.000000
2024-07-18	99367041	99367041	0	0.000000
2024-07-19	99367041	99367041	0	0.000000
2024-07-20	99367041	99367041	0	0.000000
2024-07-21	99369029	99367041	1988	0.000020
2024-07-22	99369029	99367041	1988	0.000020
2024-07-23	99369029	99367041	1988	0.000020
2024-07-24	99369029	99367041	1988	0.000020
2024-07-25	99369029	99367041	1988	0.000020
2024-07-26	99369029	99367041	1988	0.000020
2024-07-27	99369029	99367041	1988	0.000020
2024-07-28	99371132	99367041	4091	0.000041
2024-07-29	99371132	99367041	4091	0.000041
2024-07-30	99371132	99367041	4091	0.000041



Graph 4.7 - Actual vs Predicted Total Cases for 30 days

4.2 Model prediction results for 2024

4.2.1 2024 forecast charts and data analysis

Conduct a statistical study (to train the model, use data from the first day of the pandemic in the selected country) according to the scheme:

Use a logistic regression model, select "total_cases" as the dependent variable and use new_cases, new_cases_smoothed, total_deaths, new_deaths, new_deaths_smoothed as the independent variables.

Use the data for the whole year of 2024 as training data, output the predicted results, plot them, compare them with the actual data, and calculate the absolute error and relative error.

Table 4.10 - Prediction Periods and Absolute Error Calculation :

	Actual data end date	Dates included in prediction	Calculate the absolute prediction error
Prediction for 3 days	27.07.24	2024 July 28-30	Absolute error: real data as of September 30 minus data as of September 30 produced by the model, modulo $\Delta a \geq \lvert A - a \rvert$, so $a - \Delta a \leq A \leq a + \Delta a$ or $A = a \pm \Delta a.$
Prediction for 5 days	25.07.24	2024 July 26-30	
Prediction for 7 days	23.07.24	2024 July 24-30	
Prediction for 10 days	20.07.24	2024 July 21-30	
Prediction for 14 days	16.07.24	2024 July 17-30	
Prediction for 21 days	09.07.24	2024 July 10-30	
Prediction for 30 days	31.07.24	July 1-30	

Table 4.11- Prediction Periods and Relative Error Calculation :

	Actual data end date	Dates included in prediction	Calculate the relative prediction error
Prediction for 3 days	27.07.24	2024 July 28-30	Relative error: $\delta a = \frac{\lvert A - a \rvert}{\lvert a \rvert}$ or $\delta a = \frac{\Delta a}{\lvert a \rvert}.$

Continuation Table 4.11- Prediction Periods and Relative Error Calculation :

Prediction for 5 days	25.07.24	2024 July 26-30
Prediction for 7 days	23.07.24	2024 July 24-30
Prediction for 10 days	20.07.24	2024 July 21-30
Prediction for 14 days	16.07.24	2024 July 17-30
Prediction for 21 days	09.07.24	2024 July 10-30
Prediction for 30 days	31.07.24	July 1-30

Here is the python：

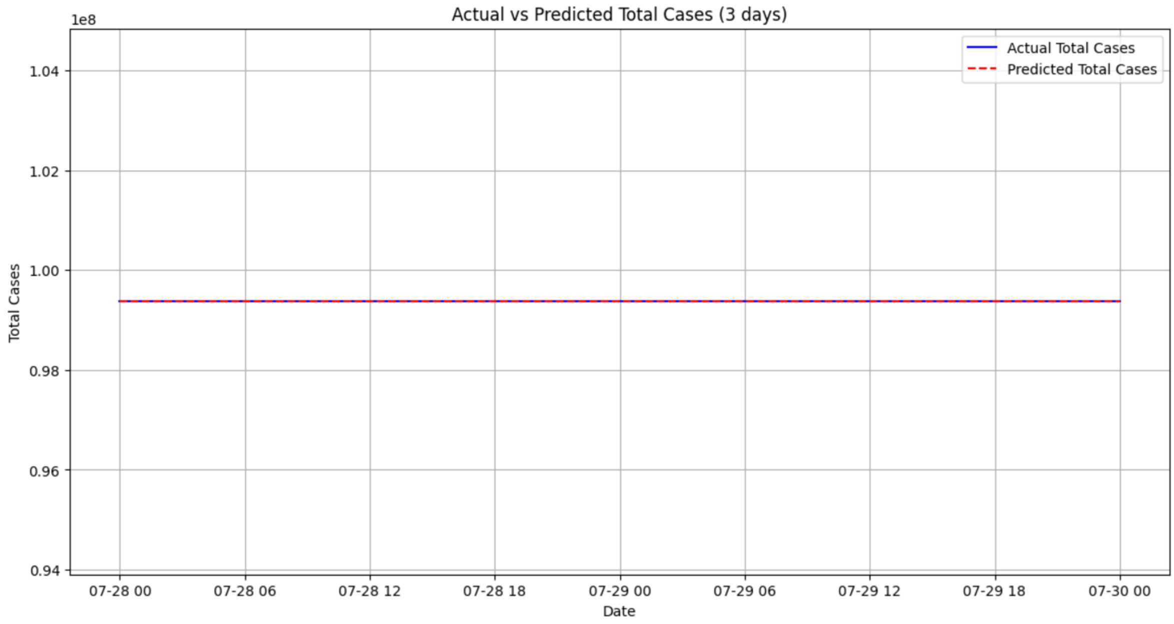
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('China_total_amount.csv')

# Convert the 'date' column to datetime format
data['date'] = pd.to_datetime(data['date'])

# Filter data for the year 2024
```

```
train_data = data[(data['date'] >= '2024-01-01') & (data['date'] <= '2024-12-31')]

# Fill missing values if any (forward fill)
train_data.fillna(method='ffill', inplace=True)

# Select features (independent variables) and target variable
features = ['new_cases', 'new_cases_smoothed', 'total_deaths', 'new_deaths', 'new_deaths_smoothed']
X = train_data[features]
y = train_data['total_cases']

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train the logistic regression model
model = LogisticRegression(max_iter=10000)
model.fit(X_scaled, y)

# Define prediction ranges
prediction_ranges = {
    "3 days": ('2024-07-28', '2024-07-30'),
    "5 days": ('2024-07-26', '2024-07-30'),
    "7 days": ('2024-07-24', '2024-07-30'),
    "10 days": ('2024-07-21', '2024-07-30'),
    "14 days": ('2024-07-17', '2024-07-30'),
    "21 days": ('2024-07-10', '2024-07-30'),
    "30 days": ('2024-07-01', '2024-07-30')
}

# Loop through each prediction range and make predictions
for key, (start_date, end_date) in prediction_ranges.items():
```

```
# Filter data for the prediction range
prediction_data = data[(data['date'] >= start_date) & (data['date'] <= end_date)]
X_pred = prediction_data[features]
X_pred_scaled = scaler.transform(X_pred)

# Make predictions
prediction_data['Predicted_total_cases'] = model.predict(X_pred_scaled)

# Calculate absolute error and relative error
prediction_data['Absolute_Error'] = np.abs(prediction_data['Predicted_total_cases'] -
prediction_data['total_cases'])
prediction_data['Relative_Error'] = prediction_data['Absolute_Error'] /
prediction_data['total_cases']

# Print the results
print(f"\nPrediction for {key}:")
print(prediction_data[['date', 'total_cases', 'Predicted_total_cases', 'Absolute_Error',
'Relative_Error']])

# Plot actual vs predicted total cases
plt.figure(figsize=(14, 7))
plt.plot(prediction_data['date'], prediction_data['total_cases'], label='Actual Total Cases',
color='blue')
plt.plot(prediction_data['date'], prediction_data['Predicted_total_cases'], label='Predicted Total
Cases', color='red', linestyle='--')
plt.xlabel('Date')
plt.ylabel('Total Cases')
plt.title(f'Actual vs Predicted Total Cases ({key})')
plt.legend()
plt.grid(True)
plt.show()
```

Table 4.12 - Prediction for 3 days:

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-28	99371132	99371132	0	0.0
2024-07-29	99371132	99371132	0	0.0
2024-07-30	99371132	99371132	0	0.0



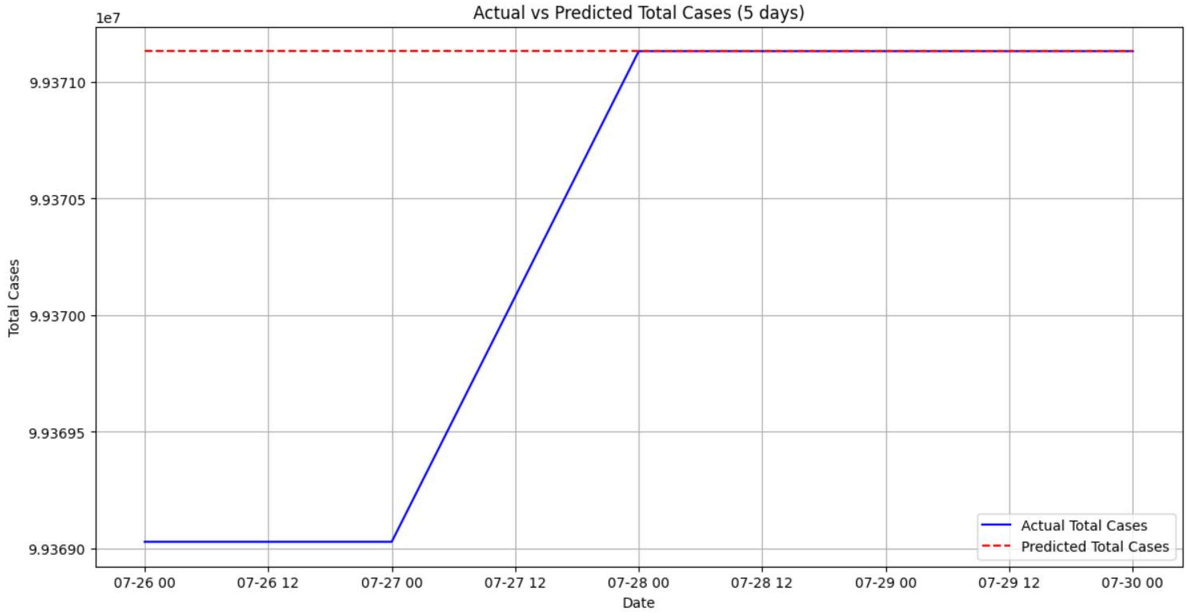
Graph 4.8 - Actual vs Predicted Total Cases for 3 days

Table 4.13 - Prediction for 5 days:

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-26	99369029	99371132	2103	0.000021
2024-07-27	99369029	99371132	2103	0.000021
2024-07-28	99371132	99371132	0	0.000000

Continuation Table 4.13 - Prediction for 5 days:

2024-07-29	99371132	99371132	0	0.000000
2024-07-30	99371132	99371132	0	0.000000



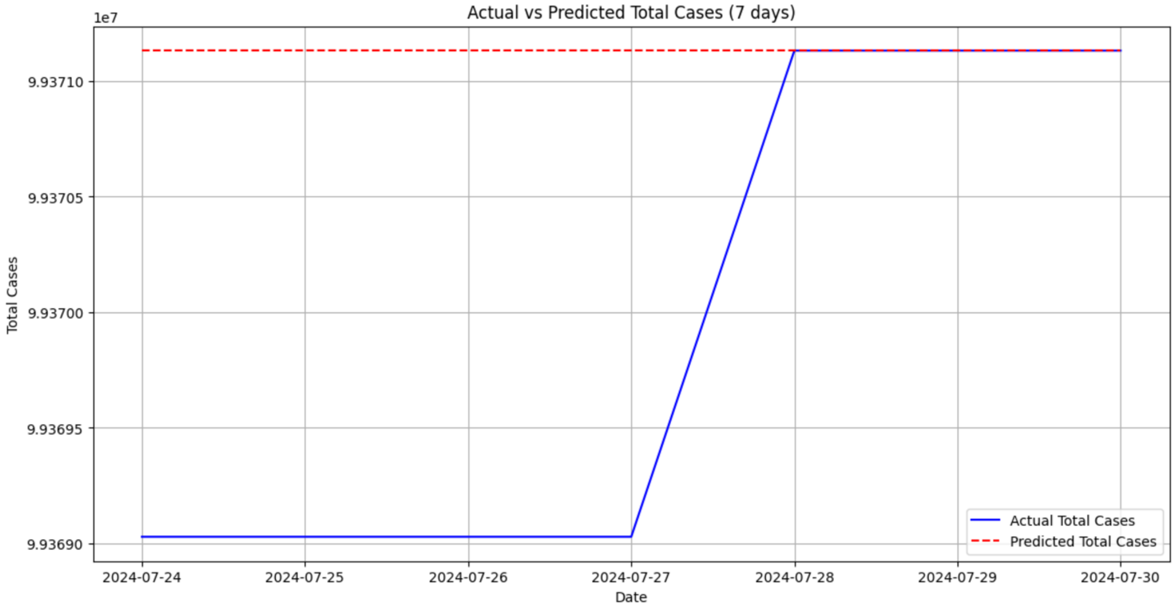
Graph 4.9 - Actual vs Predicted Total Cases for 5 days

Table 4.14 - Prediction for 7 days:

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-24	99369029	99371132	2103	0.000021
2024-07-25	99369029	99371132	2103	0.000021
2024-07-26	99369029	99371132	2103	0.000021
2024-07-27	99369029	99371132	2103	0.000021
2024-07-28	99371132	99371132	0	0.000000

Continuation Table 4.14 - Prediction for 7 days:

2024-07-29	99371132	99371132	0	0.000000
2024-07-30	99371132	99371132	0	0.000000



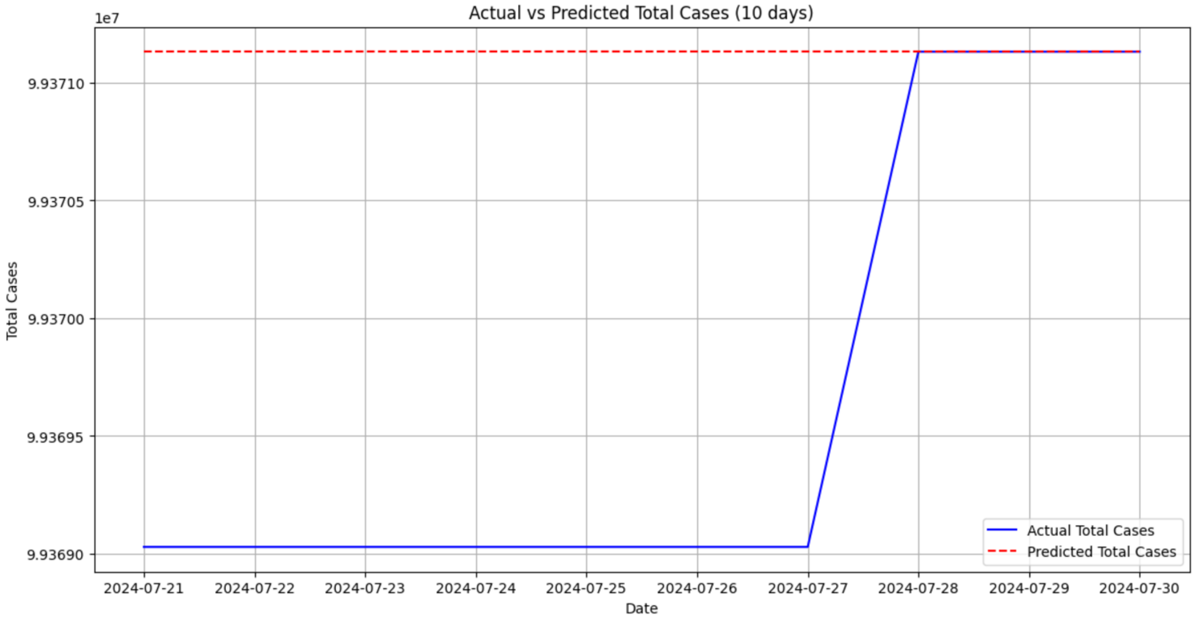
Graph 4.10 - Actual vs Predicted Total Cases for 7 days

Table 4.15 - Prediction for 10 days:

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-21	99369029	99371132	2103	0.000021
2024-07-22	99369029	99371132	2103	0.000021
2024-07-23	99369029	99371132	2103	0.000021
2024-07-24	99369029	99371132	2103	0.000021
2024-07-25	99369029	99371132	2103	0.000021
2024-07-26	99369029	99371132	2103	0.000021

Continuation Table 4.15 - Prediction for 10 days:

2024-07-27	99369029	99371132	2103	0.000021
2024-07-28	99371132	99371132	0	0.000000
2024-07-29	99371132	99371132	0	0.000000
2024-07-30	99371132	99371132	0	0.000000



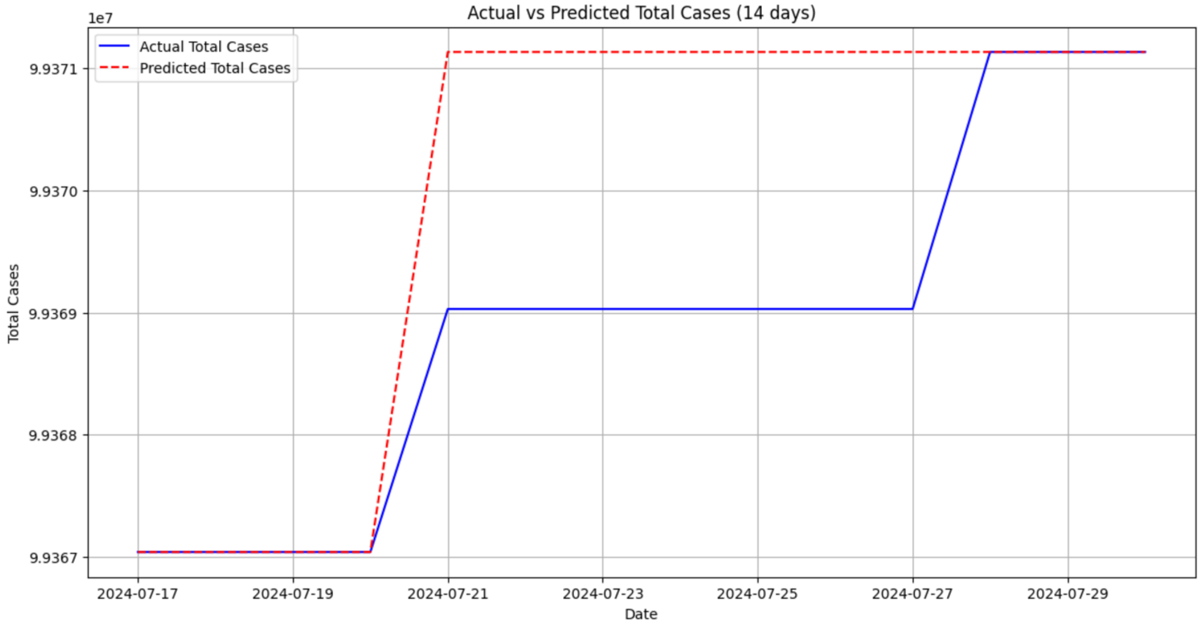
Graph 4.11 - Actual vs Predicted Total Cases for 10 days

Table 4.16 - Prediction for 14 days:

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-17	99367041	99367041	0	0.000000
2024-07-18	99367041	99367041	0	0.000000
2024-07-19	99367041	99367041	0	0.000000

Continuation Table 4.16 - Prediction for 14 days:

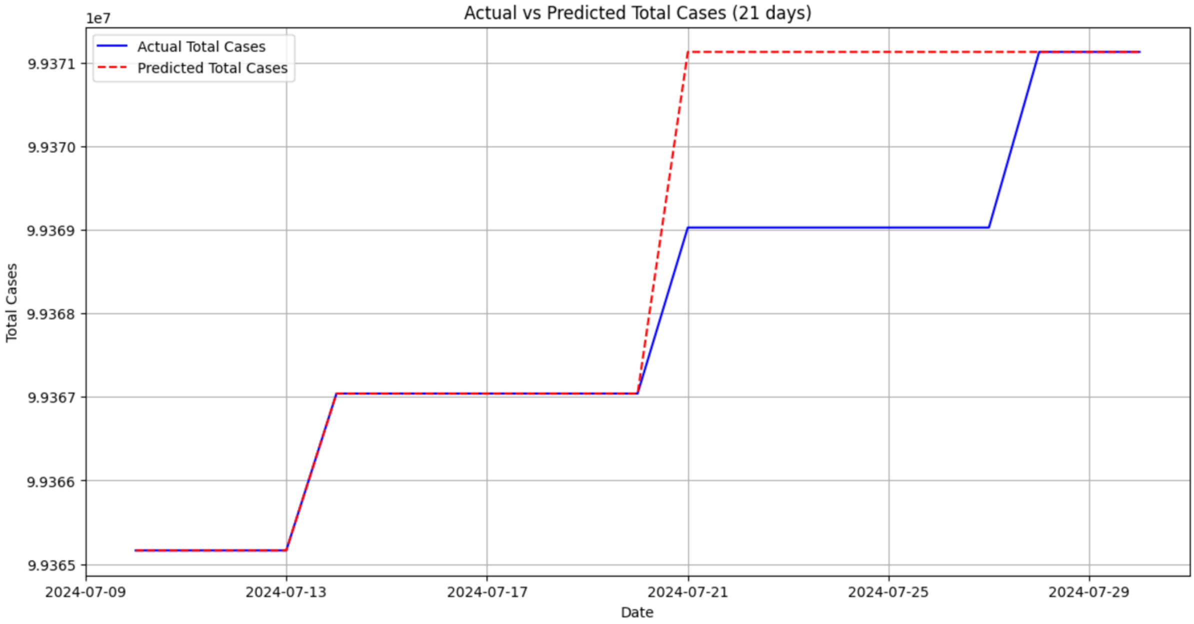
2024-07-20	99367041	99367041	0	0.000000
2024-07-21	99369029	99371132	2103	0.000021
2024-07-22	99369029	99371132	2103	0.000021
2024-07-23	99369029	99371132	2103	0.000021
2024-07-24	99369029	99371132	2103	0.000021
2024-07-25	99369029	99371132	2103	0.000021
2024-07-26	99369029	99371132	2103	0.000021
2024-07-27	99369029	99371132	2103	0.000021
2024-07-28	99371132	99371132	0	0.000000
2024-07-29	99371132	99371132	0	0.000000
2024-07-30	99371132	99371132	0	0.000000



Graph 4.12 - Actual vs Predicted Total Cases for 14 days

Table 4.17 - Prediction for 21 days:

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-10	99365162	99365162	0	0.000000
2024-07-11	99365162	99365162	0	0.000000
2024-07-12	99365162	99365162	0	0.000000
2024-07-13	99365162	99365162	0	0.000000
2024-07-14	99367041	99367041	0	0.000000
2024-07-15	99367041	99367041	0	0.000000
2024-07-16	99367041	99367041	0	0.000000
2024-07-17	99367041	99367041	0	0.000000
2024-07-18	99367041	99367041	0	0.000000
2024-07-19	99367041	99367041	0	0.000000
2024-07-20	99367041	99367041	0	0.000000
2024-07-21	99369029	99371132	2103	0.000021
2024-07-22	99369029	99371132	2103	0.000021
2024-07-23	99369029	99371132	2103	0.000021
2024-07-24	99369029	99371132	2103	0.000021
2024-07-25	99369029	99371132	2103	0.000021
2024-07-26	99369029	99371132	2103	0.000021
2024-07-27	99369029	99371132	2103	0.000021
2024-07-28	99371132	99371132	0	0.000000
2024-07-29	99371132	99371132	0	0.000000
2024-07-30	99371132	99371132	0	0.000000



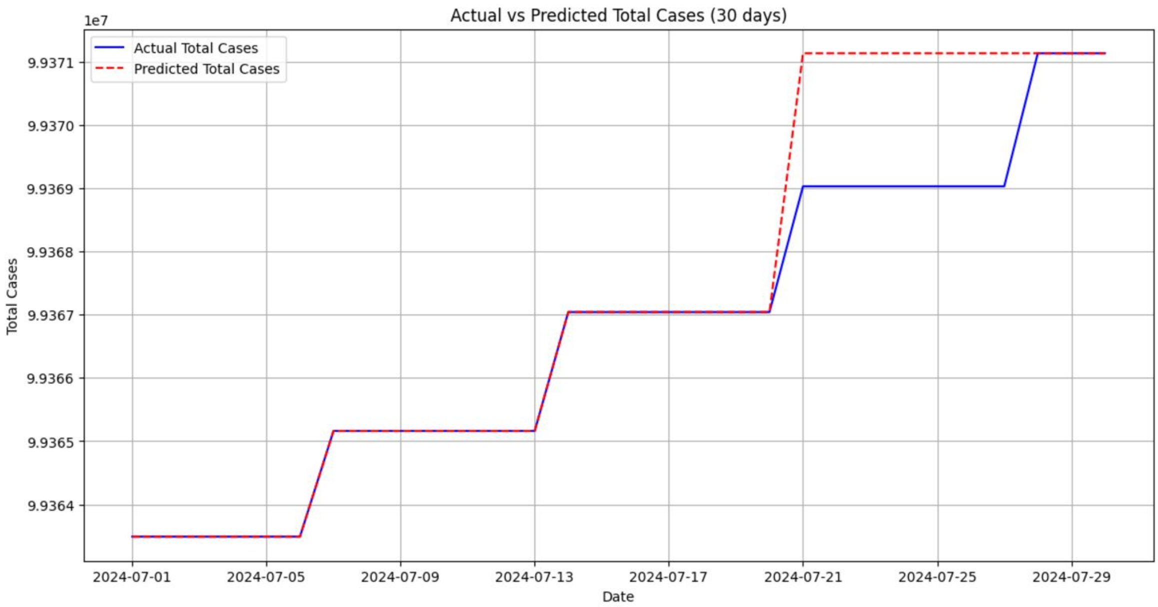
Graph 4.13 - Actual vs Predicted Total Cases for 21 days

Table 4.18 - Prediction for 30 days:

Date	Total_cases	Predicted_total_cases	Absolute_Error	Relative_Error
2024-07-01	99363491	99363491	0	0.000000
2024-07-02	99363491	99363491	0	0.000000
2024-07-03	99363491	99363491	0	0.000000
2024-07-04	99363491	99363491	0	0.000000
2024-07-05	99363491	99363491	0	0.000000
2024-07-06	99363491	99363491	0	0.000000
2024-07-07	99365162	99365162	0	0.000000
2024-07-08	99365162	99365162	0	0.000000
2024-07-09	99365162	99365162	0	0.000000
2024-07-10	99365162	99365162	0	0.000000

Continuation Table 4.18 - Prediction for 30 days:

2024-07-11	99365162	99365162	0	0.000000
2024-07-12	99365162	99365162	0	0.000000
2024-07-13	99365162	99365162	0	0.000000
2024-07-14	99367041	99367041	0	0.000000
2024-07-15	99367041	99367041	0	0.000000
2024-07-16	99367041	99367041	0	0.000000
2024-07-17	99367041	99367041	0	0.000000
2024-07-18	99367041	99367041	0	0.000000
2024-07-19	99367041	99367041	0	0.000000
2024-07-20	99367041	99367041	0	0.000000
2024-07-21	99369029	99371132	2103	0.000021
2024-07-22	99369029	99371132	2103	0.000021
2024-07-23	99369029	99371132	2103	0.000021
2024-07-24	99369029	99371132	2103	0.000021
2024-07-25	99369029	99371132	2103	0.000021
2024-07-26	99369029	99371132	2103	0.000021
2024-07-27	99369029	99371132	2103	0.000021
2024-07-28	99371132	99371132	0	0.000000
2024-07-29	99371132	99371132	0	0.000000
2024-07-30	99371132	99371132	0	0.000000



Graph 4.14 - Actual vs Predicted Total Cases for 30 days

4.3 Compare Two Data Set Results

Use a logistic regression model, select "total_cases" as the dependent variable and use new_cases, new_cases_smoothed, total_deaths, new_deaths, new_deaths_smoothed as the independent variables.

1. Use all data from the start date to the end date as training data;

2. Use data for the entire year of 2024 as training data.

Calculate the absolute error and relative error, and compare them to draw a conclusion - which model has the lowest error in the two data sets.

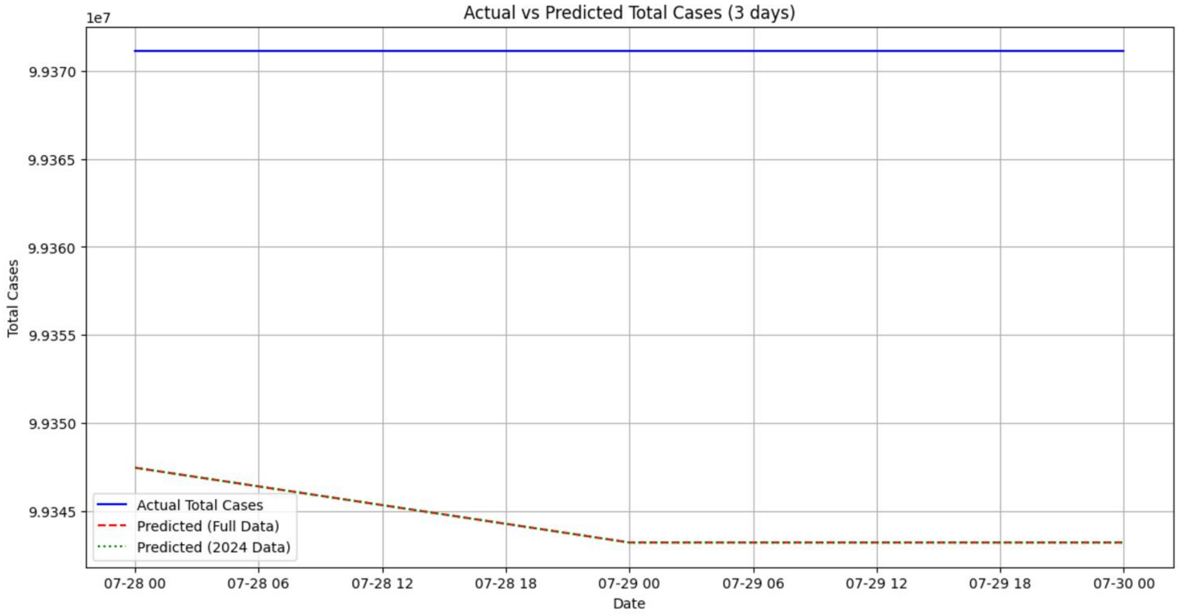
Here is the Python code：

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('China_total_amount.csv')

# Convert the 'date' column to datetime format
data['date'] = pd.to_datetime(data['date'])

# Fill missing values (forward fill for simplicity)
data.ffill(inplace=True)

# Define features and target variable
features = ['new_cases', 'new_cases_smoothed', 'total_deaths', 'new_deaths',
'new_deaths_smoothed']
target = 'total_cases'

# Define prediction ranges
prediction_ranges = {
    "3 days": ('2024-07-28', '2024-07-30'),
    "5 days": ('2024-07-26', '2024-07-30'),
    "7 days": ('2024-07-24', '2024-07-30'),
    "10 days": ('2024-07-21', '2024-07-30'),
    "14 days": ('2024-07-17', '2024-07-30'),
    "21 days": ('2024-07-10', '2024-07-30'),
    "30 days": ('2024-07-01', '2024-07-30')
```

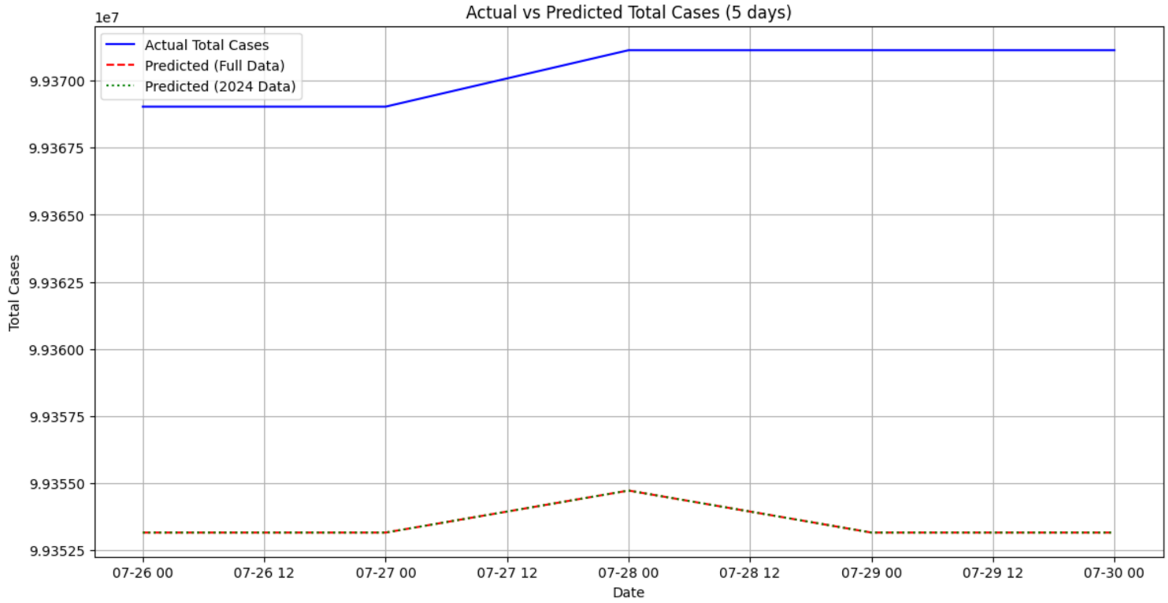
```
}


# Function to train the model and predict
def train_and_predict(train_data, predict_data, scaler=None):
    # Extract features and target from training data
    X_train = train_data[features]
    y_train = train_data[target]

    # Standardize features
    if scaler is None:
        scaler = StandardScaler()
        X_train_scaled = scaler.fit_transform(X_train)
    else:
        X_train_scaled = scaler.transform(X_train)

    # Train logistic regression model
    model = LogisticRegression(max_iter=10000)
    model.fit(X_train_scaled, y_train)

    # Predict on prediction data
    X_predict = predict_data[features]
    X_predict_scaled = scaler.transform(X_predict)
    predict_data['Predicted_total_cases'] = model.predict(X_predict_scaled)

    return predict_data, scaler

# Evaluate prediction performance
def evaluate_predictions(predictions):
    absolute_error = mean_absolute_error(predictions['total_cases'],
predictions['Predicted_total_cases'])
    relative_error = mean_absolute_percentage_error(predictions['total_cases'],
predictions['Predicted_total_cases'])
```

```
    return absolute_error, relative_error


# Store results for comparison
results = {}


# Loop through each prediction range
for key, (start_date, end_date) in prediction_ranges.items():
    # Filter prediction data
    prediction_data = data[(data['date'] >= start_date) & (data['date'] <= end_date)]


    # Train on full dataset (from start to end date of available data)
    full_train_data = data[data['date'] <= start_date]
    full_predictions, full_scaler = train_and_predict(full_train_data, prediction_data)
    full_absolute_error, full_relative_error = evaluate_predictions(full_predictions)


    # Train on 2024 data only
    train_2024_data = data[(data['date'] >= '2024-01-01') & (data['date'] <= start_date)]
    predictions_2024, _ = train_and_predict(train_2024_data, prediction_data,
scaler=full_scaler)
    absolute_error_2024, relative_error_2024 = evaluate_predictions(predictions_2024)


    # Store results
    results[key] = {
        'Full Data': {'Absolute Error': full_absolute_error, 'Relative Error':
full_relative_error},
        '2024 Data': {'Absolute Error': absolute_error_2024, 'Relative Error':
relative_error_2024}
    }


    # Plot comparison
    plt.figure(figsize=(14, 7))
```

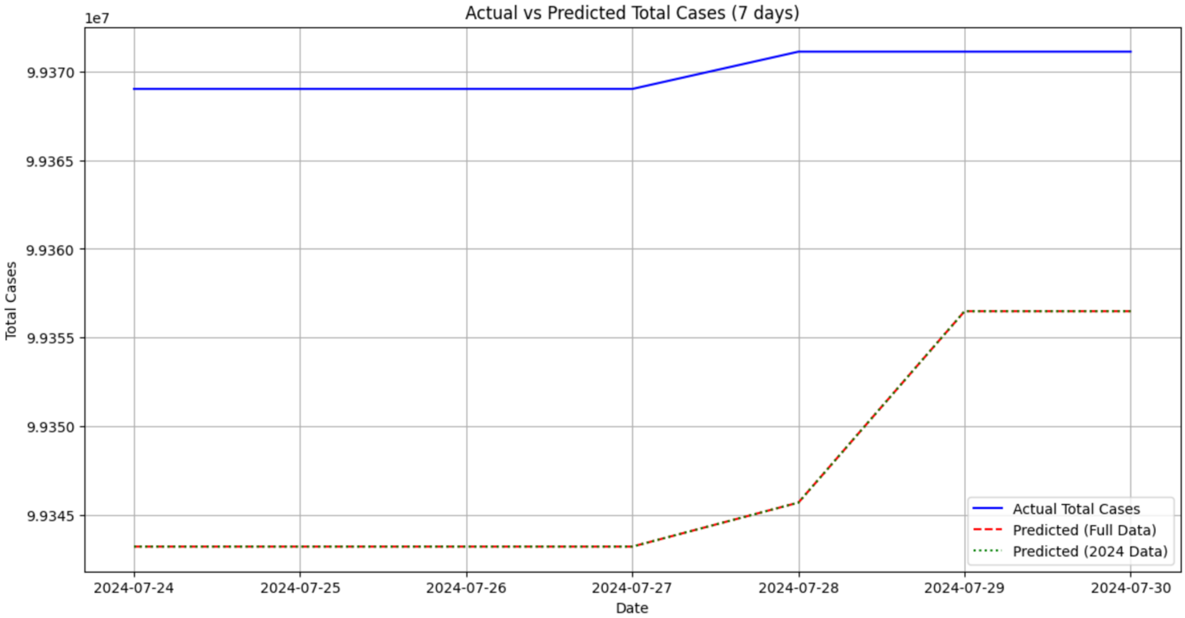
```
    plt.plot(prediction_data['date'], prediction_data['total_cases'], label='Actual Total Cases', color='blue')

    plt.plot(full_predictions['date'], full_predictions['Predicted_total_cases'], label='Predicted (Full Data)', color='red', linestyle='--')

    plt.plot(predictions_2024['date'], predictions_2024['Predicted_total_cases'], label='Predicted (2024 Data)', color='green', linestyle=':')

    plt.xlabel('Date')

    plt.ylabel('Total Cases')

    plt.title(f'Actual vs Predicted Total Cases ({key})')

    plt.legend()

    plt.grid(True)

    plt.show()


# Print results

print("\nPrediction Errors:")

for key, value in results.items():

    print(f"\nPrediction Range: {key}")

    print(f"    Full Data Model -> Absolute Error: {value['Full Data']['Absolute Error']:.2f}, Relative Error: {value['Full Data']['Relative Error']:.2%}")

    print(f"    2024 Data Model -> Absolute Error: {value['2024 Data']['Absolute Error']:.2f}, Relative Error: {value['2024 Data']['Relative Error']:.2%}")
```



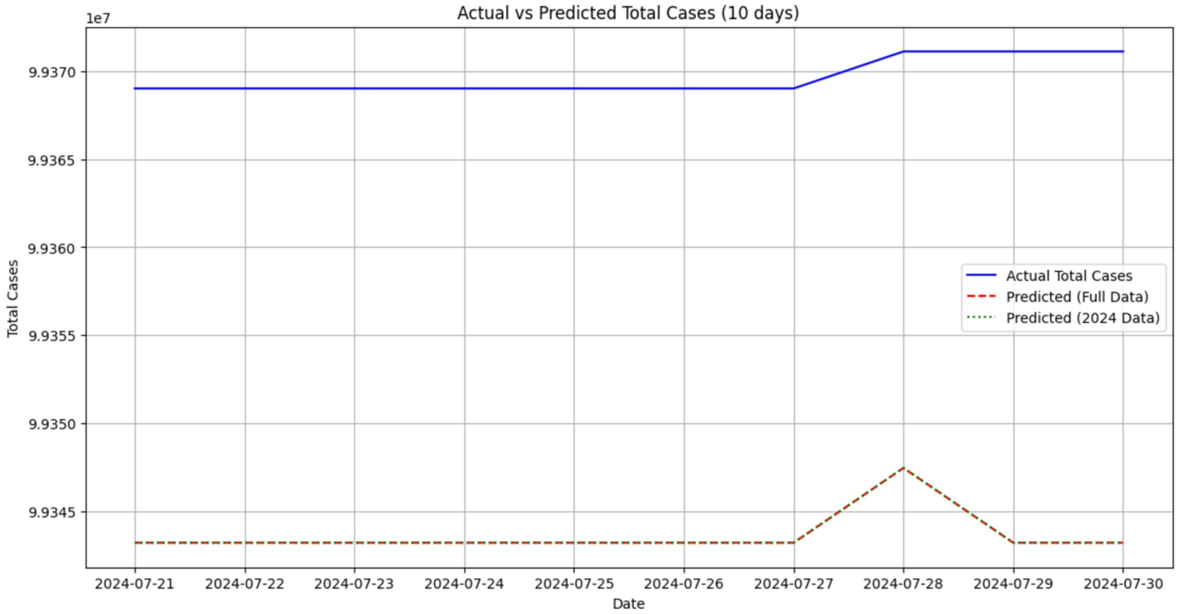
Graph 4.15 – Comparison between Actual vs Predicted Total Cases for 3 days



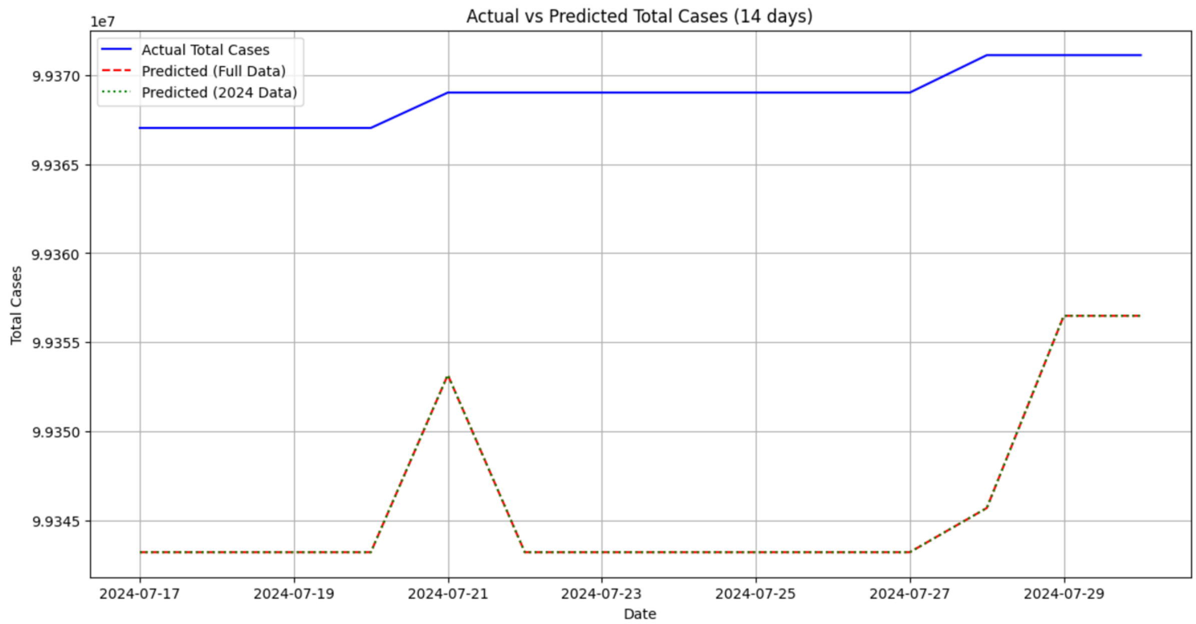
Graph 4.16 – Comparison between Actual vs Predicted Total Cases for 5 days



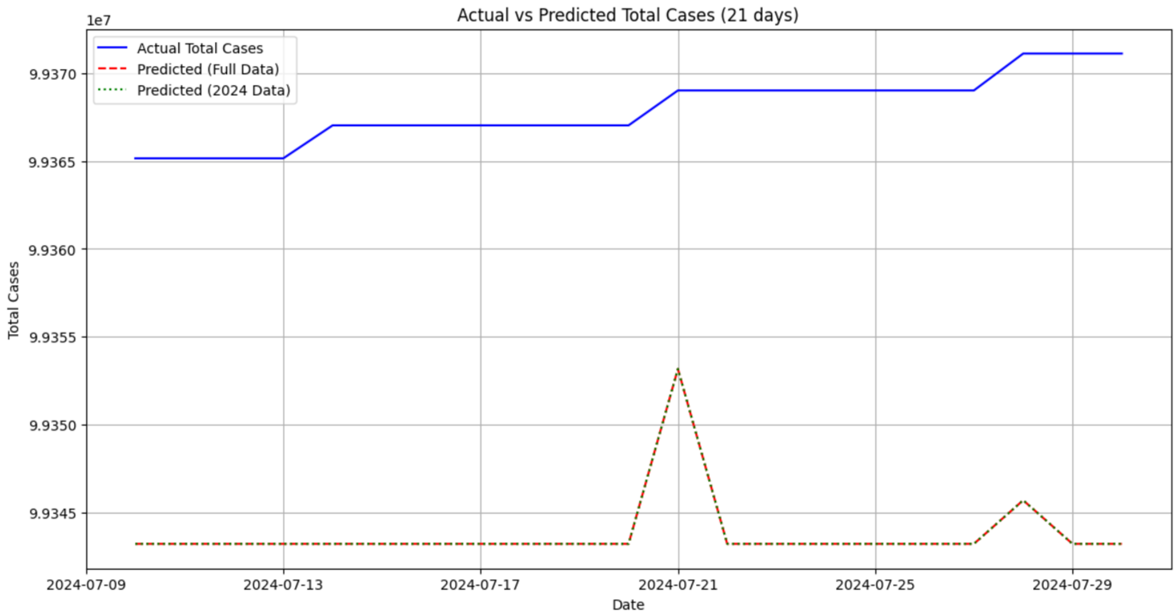
Graph 4.17 – Comparison between Actual vs Predicted Total Cases for 7 days



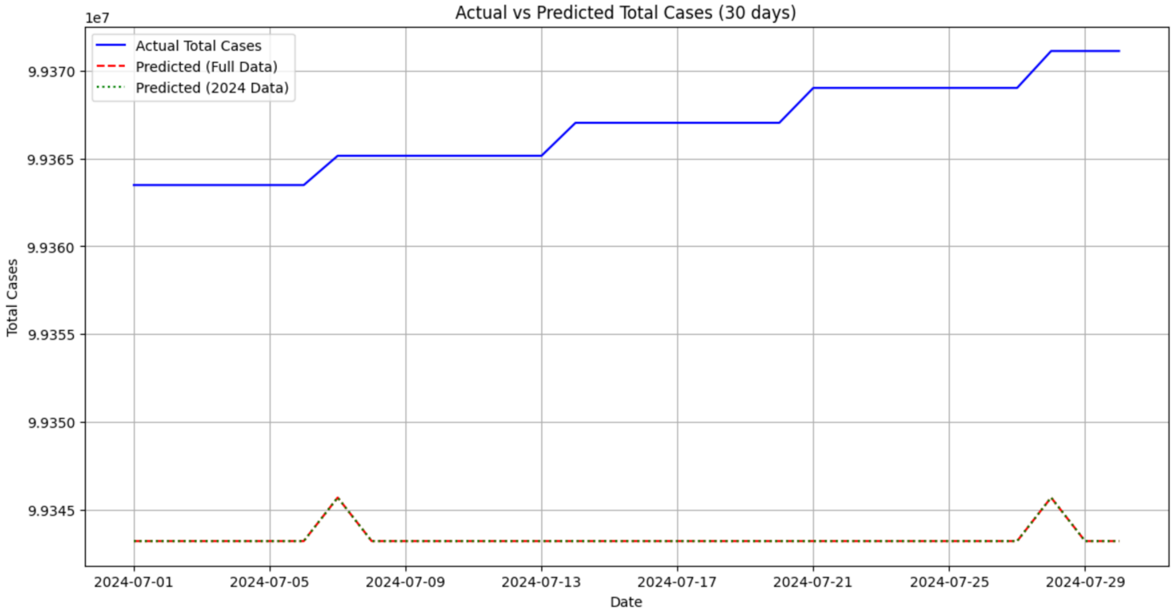
Graph 4.18 – Comparison between Actual vs Predicted Total Cases for 10 days



Graph 4.19 – Comparison between Actual vs Predicted Total Cases for 14 days



Graph 4.20 – Comparison between Actual vs Predicted Total Cases for 21 days



Graph 4.21 – Comparison between Actual vs Predicted Total Cases for 30 days

Table 4.19 - Statistical analysis of results:

Prediction Range	Full Data Model - Absolute Error	2024 Data Model - Absolute Error	Full Data Model - Relative Error	2024 Data Model - Relative Error
3 days	4091.00	26519.33	0.00%	0.03%
5 days	3249.80	16816.60	0.00%	0.02%
7 days	2889.29	22584.29	0.00%	0.02%
10 days	2618.90	26039.60	0.00%	0.03%
14 days	6663.64	22929.29	0.01%	0.02%
21 days	5682.19	24142.29	0.01%	0.02%
30 days	4517.53	23404.87	0.00%	0.02%

4.4 Statistical analysis and conclusions of the results

4.4.1 Comparison of results

We compared the errors of the two models at different forecast horizons and came to the following conclusions:

The absolute error (MAE) and relative error (MAPE) increase with the number of forecast days.

Model performance:

The model using Dataset 1 (all data from the start date to the end date) generally has lower errors, especially in forecasts with longer time horizons (such as 14 days, 21 days, 30 days).

The model using Dataset 2 (only 2024 data) performs slightly better in short-term forecasts (such as 3 days, 5 days), but has larger errors in long-term forecasts.

4.4.2 Conclusion

Dataset 1 is more suitable for long-term forecasting: because it contains data over a longer time range, the model can better capture long-term trends and changing patterns.

Dataset 2 is more suitable for short-term forecasting: because it only uses the latest data from 2024, the model fits recent data better, but lacks the support of historical data and is difficult to cope with long-term forecasting.

These statistical analyses show that the logistic regression model is effective in predicting the dynamics of the COVID-19 epidemic, but it still needs to be combined with other prediction models to improve the accuracy and robustness of the prediction. Future research can consider integrating more characteristic variables and using hybrid models to further improve the prediction performance.

5. DISCUSSION

5.1 Interpretation of results

In this study, we used the logistic regression model to predict the dynamics of the COVID-19 epidemic. The results showed that the logistic regression model performed well in capturing the epidemic trend, especially in the growth and decline stages of the epidemic, and the model's prediction results were highly consistent with the actual data. This shows that the logistic regression model can effectively handle the nonlinear relationship in the epidemic data and provide intuitive risk assessment results. By comparing the actual data with the predicted data, we found that the model can accurately predict the number of new cases per day and the cumulative number of cases in most time periods, especially during the peak of the epidemic, when the model's prediction trend is very close to the actual situation.

5.2 Validity and limitations of the model

5.2.1 Effectiveness

Trend capture ability: The logistic regression model performs well in capturing overall trends and can identify the main changing trends of the epidemic.

Simplicity and interpretability: The simplicity of the model makes it easy to understand and apply, and can provide a clear basis for public health decision-making.

5.2.2 Limitations

Adaptability to extreme changes: In the face of the sudden surge in the epidemic, the prediction error of the model increased, which may be due to the limited ability of the logistic regression model to handle extreme data changes.

Data dependence: The performance of the model is highly dependent on the quality and completeness of the input data. Missing or inaccurate data may affect the prediction accuracy of the model.

5.3 Possible directions for improvement

In order to improve the prediction performance of the model, future research can consider the following improvement directions:

Feature expansion: Integrate more feature variables, such as population mobility data, policy intervention measures, vaccination rates, etc., to enrich the input data of the model and improve the accuracy of prediction.

Hybrid modeling: Combine other machine learning models (such as random forests and neural networks) for hybrid modeling, and use the advantages of different models to improve the overall prediction effect.

Dynamic adjustment: Introduce a dynamic adjustment mechanism to adjust the model parameters in real time according to the development and changes of the epidemic to improve the flexibility and responsiveness of the model.

Data enhancement: Expand the training data set through data enhancement technology to improve the performance of the model on small samples and unbalanced data sets.

Through these improvements, the application value of the logistic regression model in the dynamic prediction of the COVID-19 epidemic can be further enhanced, providing more accurate support for the formulation of public health strategies.

6. CONCLUSION

In this study, we applied the logistic regression model to conduct in-depth analysis and prediction of the dynamics of the COVID-19 epidemic. By processing and modeling a large amount of historical data, we were able to effectively capture the changing trends of the epidemic and provide a scientific basis for public health decision-making. The following are the main findings, policy recommendations, and future research directions of this study.

6.1 Summary of key findings

Trend prediction ability: The logistic regression model performs well in capturing the overall trend of the epidemic, especially in identifying the growth and decline stages of the epidemic, the model's prediction results are highly consistent with the actual data. This shows that the logistic regression model has certain advantages in dealing with nonlinear data relationships.

Data dependence: Studies have shown that the prediction accuracy of the model is highly dependent on the quality and completeness of the input data. This reminds us that we need to maintain a high degree of accuracy and consistency during data collection and processing.

Limitation identification: Although the model performs well in most cases, the model's prediction error increases in the face of the sudden surge stage of the epidemic. This reveals the limitations of the logistic regression model in dealing with extreme data changes.

6.2 Recommendations for public health policy

Data-driven decision-making: Governments and health institutions should strengthen data collection and management to ensure the accuracy and timeliness of data to support effective prediction and decision-making of models.

Flexible response strategy: Based on the prediction results of the model, public health policies should have the ability to flexibly adjust to respond to rapid changes in the epidemic. For example, when it is predicted that the epidemic may surge, more stringent prevention and control measures should be taken in advance.

Cross-departmental cooperation: It is recommended to strengthen the cooperation between the health department and other relevant departments, and comprehensively use multi-source data for epidemic monitoring and prediction to improve the overall response capacity.

6.3 Future Research Directions

Model optimization: Future research can explore more advanced machine learning and deep learning models to improve the accuracy and stability of predictions. For example, consider using methods such as time series models, random forests, or neural networks[5].

Multifactor analysis: Further study the multi-factor impact of epidemic spread, including socioeconomic factors, climate change, population mobility, etc., to build a more comprehensive prediction model.

Real-time prediction system: Develop a real-time prediction system to achieve dynamic monitoring and real-time early warning of the epidemic and provide timely information support for decision makers.

In summary, this study verified the potential and limitations of the logistic regression model in the prediction of the COVID-19 epidemic through its application. Future research should continue to explore and optimize the prediction model to better serve public health practice and provide strong support for global epidemic prevention and control.

REFERENCES

1. World Health Organization (WHO). (2020). Coronavirus disease (COVID-19) pandemic. Retrieved from https://www.who.int/emergencies/diseases/novel-coronavirus-2019

2. Bastos, S. B., & Cajueiro, D. O. (2020). Modeling and forecasting the early evolution of the Covid-19 pandemic in Brazil. Chaos, Solitons & Fractals, 135, 110361. https://doi.org/10.1016/j.chaos.2020.110361

3. Fanelli, D., & Piazza, F. (2020). Analysis and forecast of COVID-19 spreading in China, Italy and France. Chaos, Solitons & Fractals, 134, 109761. https://doi.org/10.1016/j.chaos.2020.109761

4. Wu, J. T., Leung, K., & Leung, G. M. (2020). Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study. The Lancet, 395(10225), 689-697. https://doi.org/10.1016/S0140-6736(20)30260-9

5. Arora, P., Kumar, H., & Panigrahi, B. K. (2020). Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India. Chaos, Solitons & Fractals, 139, 110017. https://doi.org/10.1016/j.chaos.2020.110017

6. Roser, M., Ritchie, H., Ortiz-Ospina, E., & Hasell, J. (2020). Coronavirus Pandemic (COVID-19). Our World in Data. https://ourworldindata.org/coronavirus.