**Background**

In 2022, when the novel coronavirus pandemic was rampant, a puzzle game called Wordle, which was run on the platform of Twitter, attracted a lot of attention in a short time and then became popular all over the world. By January 2023, more than 300,000 people were taking part daily, according to the New York Times.

The rules of the game are pretty simple: You try to solve a puzzle by guessing a five-letter word six or fewer times, and you get feedback for each guess: if you have the right letter in the right spot, it shows up green. A correct letter in the wrong spot shows up yellow. A letter that isn't in the word in any spot shows up gray. For this version, each guess must be an actual English word, and guesses that are not recognized as words by the contest are not allowed.

For the New York Times staff overseeing the game, they desperately needed a way to do statistical analysis of the data from past puzzles, and a way to assess the difficulty of the puzzles chosen that day.

**Restatement**

To optimize The Times' evaluation system, MCM produced a daily results file from Jan. 7 to Dec. 31, 2022. We will complete the following tasks according to the given data:

1. Develop a model to account for daily variations in the number of reported results and explain whether word attributes affect the percentage of reported scores.

2. Build a model to predict the relevant percentage of one day in the future (1,2,3,4,5,6,X), and then evaluate the model by analyzing the performance of the training set

3. Establish a model that can distinguish the difficulty of words, and analyze the accuracy of the model

4. Perform descriptive analysis or other statistical analysis on the data set

**Description of the ARIMA model**

**ARIMA**(Autoregressive Integrated Moving Average mode) was a famous time series prediction model proposed by Box and Jenkins in the 1970s. The basic idea of ARIMA model is to treat the data sequence formed by the prediction object over time as a random sequence, which is approximated by a certain mathematical model. Once it identified, the model can predict future values from past and present values of the time series.

The model can be divided into three parts. The bottom layer is the AR model, which uses the same variable such as the previous periods of x, namelyto, to predict the performance ofin the current period, and assumes that they are linear relationships. The formula is:

Where c is a constant term; is assumed to be the random error of the mean equal to 0 and the standard deviation equal to σ (zero mean white noise sequence); Sigma is assumed to be invariant for any t.

On this basis, we add **MA** model to optimize **AR** model. **MA** model focuses on the accumulation of error terms in the autoregressive model, which can effectively eliminate random fluctuations in the prediction. Its formula is defined as

However, due to the limitation that AR model requires time series to have stationarity, we still need to continue to optimize it

For a non-stationary time series, after the elimination of its local level or trend, it will show a certain homogeneity, in other words, some parts of the series are very similar to others. This kind of non-stationary time series can be transformed into stationary time series after difference processing, which is called homogeneous non-stationary time series, in which the degree of difference is the order of homogeneity.

If I take ∇ as the difference operator, then we have

For the delay operator B, there is

Therefore, it can be concluded that

Given a homogeneous non-stationary time series of order d, then given that is a stationary time series, it can be called the ARMA(p,q) model, that is

In it

Are respectively autoregressive coefficient polynomial and moving average coefficient polynomial. is zero mean white noise sequence. The proposed model can be called the autoregressive summation moving average model, denoted as **ARIMA(p,d,q)**.

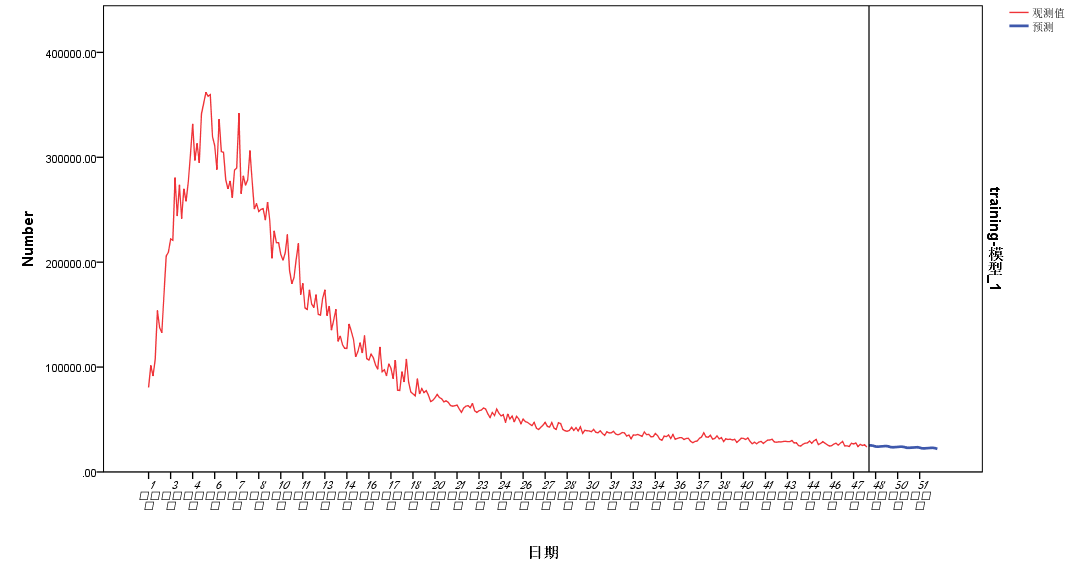
We grouped the 359 data sets we had (the data as of November 30 had been cleaned up) and used the data before November 30 as the training set and the rest as the test set.

We used IBM SPSS Statistic 20 to establish the ARIMA model, and the following results were obtained:

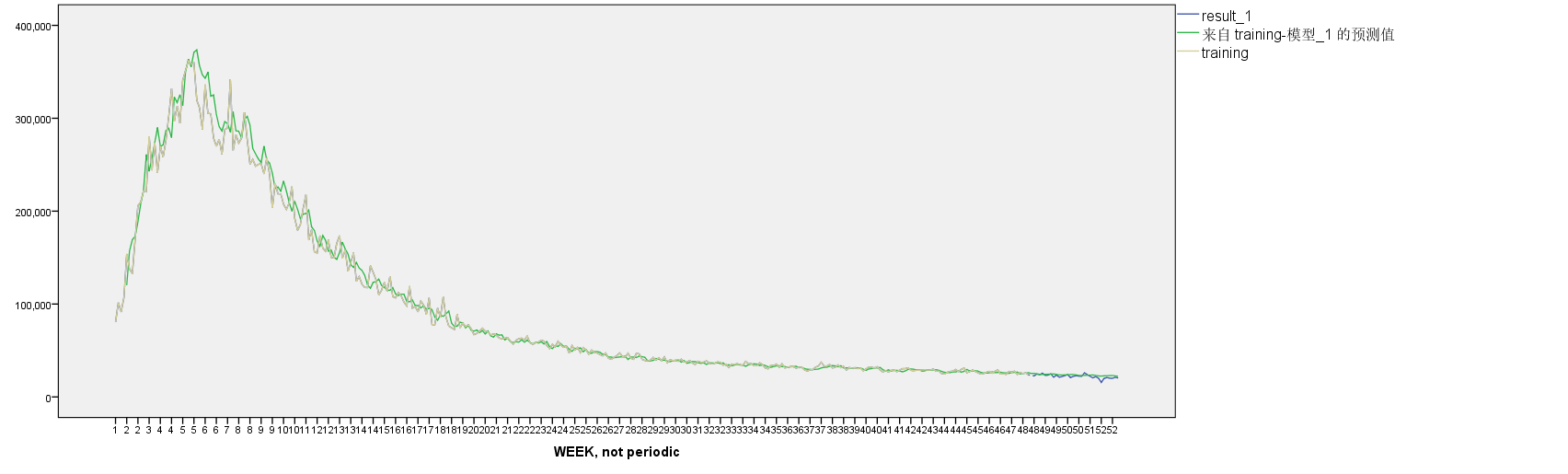
|  |  |
| --- | --- |
| Fitting statistics | mean value |
| Stationary | .277 |
|  | .981 |
| RMSE | 12846.088 |
| MAPE | 5.929 |
| Max APE | 27.579 |
| MAE | 6775.711 |
| Max AE | 58559.606 |
| normalized BIC | 19.029 |

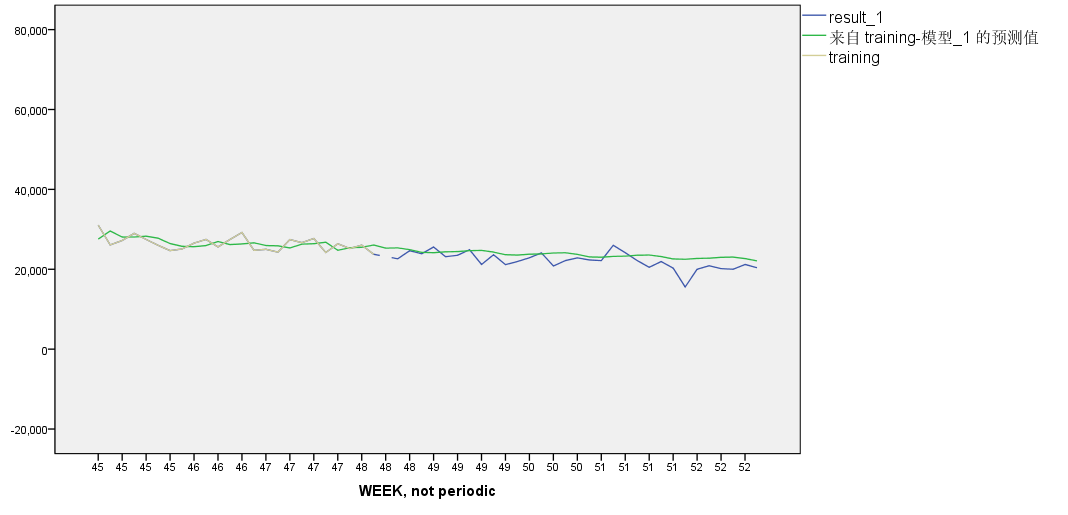
The model produces eight pieces of data above, (Variance), RMSE (Residual Mean Square), MAPE (Mean Absolute Percentage Error), MAXAPE (Maximum absolute percentage error), MAE (Mean Absolute Deviation), MAXAE (Maximum Absolute Error), normalized BIC (Bayesian Information Criterion). They describe the deviation between predicted and actual values in different ways.

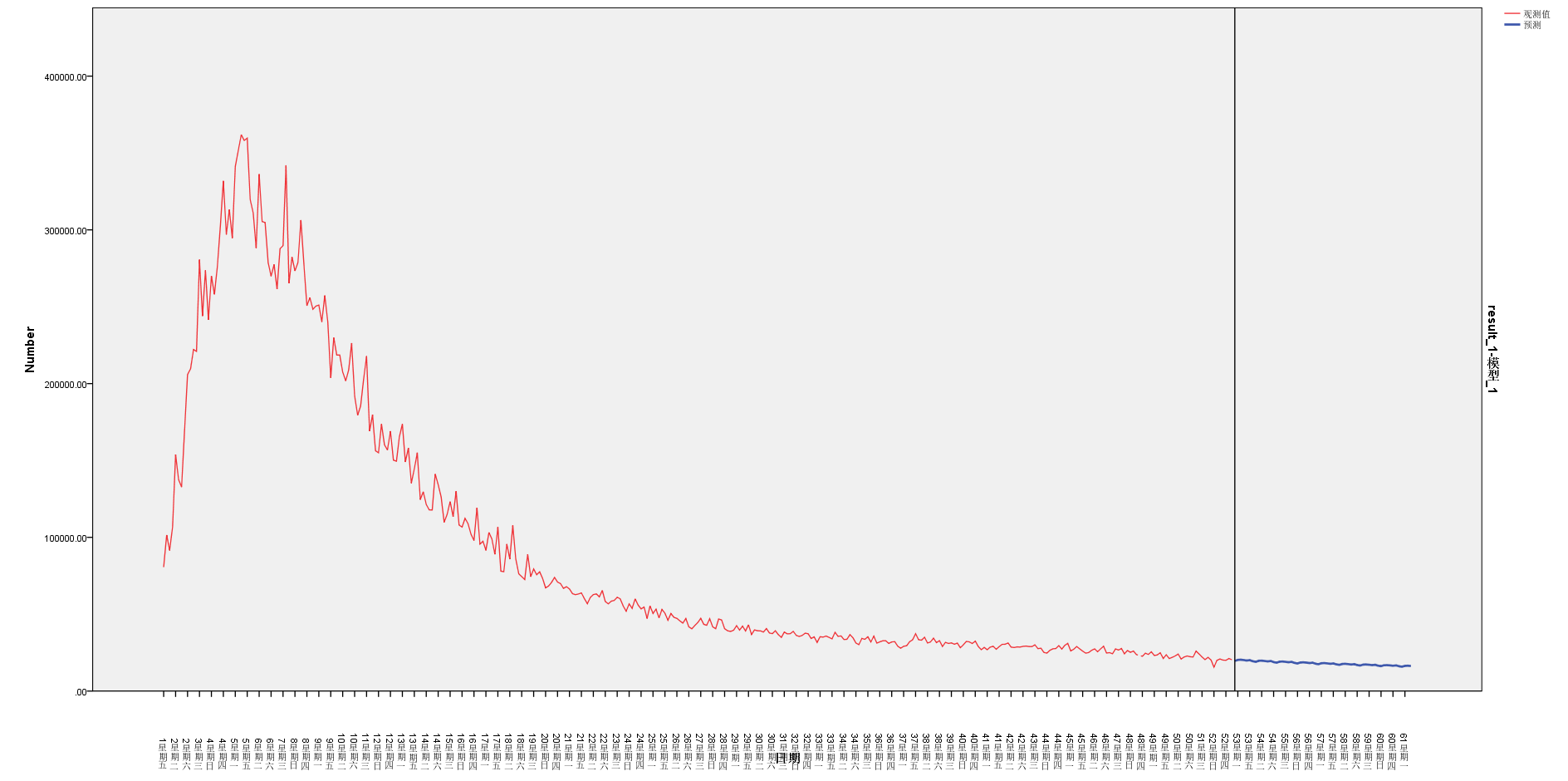
We took 328 pieces of data before November 30 as the training set to test our model, and the test results are shown in the Figure 3.



The images of the real data are generated and compared with the images of the training set



Zoom in at the end of the image where the prediction is made

The results are shown in the figure above. We can obviously see that the results of the predicted value of the training set coincide with the real data to a high degree, which indicates that our model has high reliability and accuracy.

Apply the model to the problem we need to solve, and after importing all the data sets, forecast the data after December 31, 2022, until March 1, 2023. Based on our model projections, we conclude that the range of reports for March 1, 2023 is:**（）**

**BP neural network:**

BP neural network is a multi-layer feedforward neural network trained according to the error backward propagation algorithm. The propagation process can be divided into forward propagation process and back propagation process. In the forward propagation process, BP neural network processes data through hidden layer neurons to output corresponding results. In the back propagation process, BP neural network constantly adjusts the parameter values of neurons in each layer by comparing the error between the real result and the predicted result. Thus, the error can be reduced to achieve the ideal effect of the model. The principle of adjustment is that the error is decreasing, so the weight adjustment should be proportional to the error gradient.

Assume that the number of nodes in the input layer is, the number of nodes in the hidden layer is , and the number of nodes in the output layer is . The weight from input layer to hidden layer is, the weight from hidden layer to output layer is , the bias from input layer to hidden layer is , the bias from hidden layer to output layer is , the learning rate is and the excitation function is takes the Sigmoid function, of the form as

As shown in the BP network above, the output of the hidden layer is

The output of the output layer is

In view of the error, we still need to set out the equation to take the error

Where is the expected output. If we call , the error can be denoted as

In the above formula,

The updated formula of weight is

The updating formula of bias is:

Finally, determine whether the iteration is over.

**Model establishment:**