Team Control Number

202011252167

Problem Chosen

A

2020

ShuWei Cup

Summary Sheet

Summary

China is the country with the largest steel production in the world, and rebar is one of the largest steel products in China. Rebar is an indispensable structural material for infrastructure construction and is widely used in civil engineering construction of houses, bridges, and roads. Therefore, it is of great significance to reasonably and effectively grasp the market demand dynamics of rebar. This article analyzes the demand for rebar, refers to the attached data and consults related materials, constructs the corresponding mathematical model, and makes accurate and scientific forecasts.

In the first question, in order to filter out the factors affecting the demand for rebar with a higher degree of correlation, we first clean and preprocess the data, and then resample. We adopted the gray correlation analysis model to analyze the correlation degree between the various factors that may affect the demand for rebar and the demand for rebar, and draw the correlation coefficient matrix heat map.

In the second question, in order to realize the forecast of the demand for rebar, we introduced the ARIMR model for time series data. However, there are too many influencing factors in real life, and accurate forecasting cannot be achieved by relying on the ARIMA model alone, so the gray neural network and gray prediction are introduced. For complex things, the model analyzes and processes small samples to make predictions. With the nonlinear fitting of the BP neural network and the extremely strong function approximation function, it can achieve lower errors. Finally, we build a more powerful forecasting model through combined forecasting and the weighted average method of inverse error.

In the third question, with regard to the lag in data update, we adopt the method to restore the real time series and substitute it into the combined forecasting model.

Finally, this article makes a pertinent evaluation of the model and gives some suggestions for the predicted demand for rebar.

Key word : **Grey Relational Analysis; ARIMA; Grey Neural Network; Combination Forecast**

Content

Content

1. Introduction

1.1Background

China is the world's largest producer and consumer of steel. Rebar is one of the largest steel products. Since the Shanghai Futures Exchange officially launched rebar futures in March 2009, the number of transactions has been huge, showing strong liquidity. Nearly 17 trillion yuan of rebar futures were traded on the Shanghai Futures Exchange in 2019. Steel is widely used in building, bridge, road and other civil engineering construction. It is an essential structural material for infrastructure construction. In recent years, with the rapid development of economy, China's demand for steel is further expanded, so it is of great significance to grasp the main indicators that affect the steel demand and to reasonably and effectively grasp the market demand dynamics of rebar. Forecasting the demand for rebar from the perspective of national macro-control is conducive to deepening the supply-side structural reform in the steel industry, improving the supply and demand situation, and alleviating the situation of overcapacity in the steel industry. From the perspective of commodity trading, the investment strategy of rebar futures can be adjusted according to the predicted results.

1.2Work

We need to set up a correct analysis model and to determine the influence degree of various factors on the rebar demand, on the basis of the demand for rebar according to eight of the most important indicators to establish the forecast model of the right, as a result of the release time lag, we also need to optimize the forecasting model, to make it more close to the actual application scenario.

First of all, we choose some influence indicators with complete data and high correlation by ourselves, verify the correlation by using grey correlation analysis, and then eliminate the influence variables with poor correlation by using thermal diagram. Then, the demand of rebar is predicted by ARIMA time series model and gray neural network model. Finally, we optimize the model by solving the time lag problem.

2. Problem analysis

2.1Analysis of question one

There are many factors affecting the demand of rebar. From all the influencing indicators given, we have selected 12 indicators with complete data and high correlation: Rebar Price, Steel Production, Fin Revenue, Fin Expenditure, the Construction Area, Development Funds, Steel Volume, Cement Rate, the PPP, Cement Price Index, PMI, and Unsold Area, using the grey correlation analysis to verify the correlation between them and the Rebar demand, After that, the influence variables with poor correlation are eliminated by thermal force diagram, and finally eight influencing indicators are selected, namely, Rebar Price(X1), Steel Production(X2), Fin Revenue(X3), Construction Area(X4),Steel Volume(X5), Cement Rate(X6), PPP(X7) and PMI(X8).

2.2Analysis of question two

We ignored the inconsistency between data annotation time and data update time, and took the time of data release as the default time of data label. Then we predicted and analyzed the demand of rebar through ARIMA time series model and gray neural network model.

2.3Analysis of question three

In practice, there is a lag time between the data update time and the data annotation time. For example, most monthly data is marked on the last day of each month, and data is not released until the middle of the next month. Therefore, we corrected the data to make the data the real data of data update time, optimized the model, and made the prediction again, so as to make the prediction result more consistent with the actual operation situation.

3. Symbol and Assumptions

3.1 Symbol Description

Symbol	Meaning
X1	Rebar Price
X2	Steel Production
X3	Fin Revenue
X4	Construction Area
X5	Steel Volume
X6	Cement Rate
X7	PPP
X8	PMI
Y	Rebar Demand
x^*	the processed data
x	the original data
x_{mean}	the sample average
$x_{\max}$	the maximum value of the sample
$x_{\min}$	the minimum value of the sample
ρ	resolution coefficient
r_i	the correlation degree
ε_t	the zero mean white noise sequence
L	Range of values
Q_n	The share of the nth factor

Out	output value
In	input value
w_{ij}	The weight of factors in row i and column j
Sigmoid	$f(x)=\frac{1}{1+e^{x}}$
tanh	$\frac{1-e^{-2x}}{1+e^{-2x}}$

3.2 Fundamental Assumption

3.2.1 Assumptions of question one

1.The model selects the correct variables.
2. The model selects the correct functional form.
3. There is no correlation between homogeneity of variance and sequence.
4. Random interference terms are normally distributed with zero mean value and hom -ovariance.
5.The data we have collected are completely true.

3.2.2 Assumptions of question two

1.There are only twenty-eight days in February, not the twenty-nine of February.
2. In the five years from 2016 to 2020, the rebar market was relatively stable and there was no big fluctuation.
3. There is no epidemic.
4. In the five years from 2016 to 2020, there was no economic crisis.
5.The materials we use (including computer software, the Internet, etc.) are completely reliable.

4. Model

4.1 Model of question one

4.1.1 Data filtering and preprocessing

Rebar is an indispensable structural material for infrastructure construction. There are many factors that affect the market demand for rebar, and there are also differences between the influencing factors and the impact mechanism of rebar demand. In order to accurately predict the demand for rebar, it is necessary to screen suitable variables for modeling and forecasting.

After discussion, we screened according to the degree of influence of the influencing factors on the rebar combined with the data frequency of the given data. The data frequency is uniformly weekly. The monthly influencing factors are filtered from the data, and then resampled on a weekly basis. Resampling refers to the process of converting a time series from one frequency to another.

After screening, We have selected the data of the eight influencing factors of rebar_price, steel_production, finRevenue, constructionArea, steelVolume, Cement Rate, PPP, PMI from March 6, 2016 to September 6, 2020.

4.1.2 The establishment of grey relational analysis model

By observing the filtered data, we found that our goal is to find the correlation between the dependent variable (rebar demand) and several influencing factors. To this end, we may wish to consider the use of Grey Association Analysis Model (GRA). It can analyze the degree of influence of various factors on the results, and can also solve comprehensive evaluation problems that change over time [1] , and the core is to establish a parent sequence that changes over time according to certain rules, take the change of each evaluation object over time as a subsequence, find the degree of correlation between each subsequence and the parent sequence, and draw conclusions based on the magnitude of the correlation.

The establishment of GRA model can follow the steps below:

1.Determine the reference series (parent series) $Y = Y(k) \mid k = 1,2,...n$ and comparison series (subsequence) $X_i = X_i(k) \mid k = 1,2,...m$.

2. Process variables dimensionlessly.

$$x^* = \frac{x - x_{mean}}{x_{\max} - x_{\min}}$$

Among them, x^* is the processed data, x is the original data, x_{mean} is the sample average, $x_{\max}$ is the maximum value of the sample, and $x_{\min}$ is the minimum value of the sample.

3.Calculate the correlation coefficient.

$$\xi_i(k) = \frac{i_{\min} k_{\min} \mid y(k) - x_i(k) \mid + \rho i_{\max} k_{\max} \mid y(k) - x_i(k) \mid}{\mid y(k) - x_i(k) \mid + \rho i_{\max} k_{\max} \mid y(k) - x_i(k) \mid}$$

Set $\Delta_i(k) = \mid y(k) - x_i(k) \mid$, then the above formula can be changed to

$$\xi_i(k) = \frac{i_{\min} k_{\max} \Delta_i(k) + \rho i_{\max} k_{\max} \Delta_i(k)}{\Delta_i(k) + \rho i_{\max} k_{\max} \Delta_i(k)}$$

Among them, $\rho \in (0, \infty)$, becomes the resolution coefficient. The smaller the value of ρ is, the greater the resolution is. The general value interval of ρ is (0,1), and the specific value depends on the situation. At that time when $\rho \le 0.5463$, the resolution is the best, usually makes ρ equals 0.5.

4.Calculate the degree of relevance.

Because the correlation coefficient is the value of the correlation between the comparison series and the reference series at each time (that is, each point in the curve), it has more than one number, and the information is too scattered for overall comparison. Therefore, it is necessary to concentrate the correlation coefficients at each moment (that is, each point in the curve) into one value, that is, to find the average value, as the quantitative expression of the degree of correlation between the comparison series and the reference series, the correlation degree r_i formula is as follows:

$$\mathrm{r}_i = \frac{1}{n}\sum_{k=1}^{n}\xi_i(k), k = 1,2,...n$$

We use Python to analyze the GRA model according to the above formula, and get the correlation between 12 variables (Rebar Price, Steel Production, Fin Revenue, Fin Expenditure, Construction Area, Development Funds, Steel Volume, Cement Rate, PPP, Cement Price Index, PMI, Unsold Area). Use the heat map to visualize these correlations, and get the result shown in Figure 4-1.

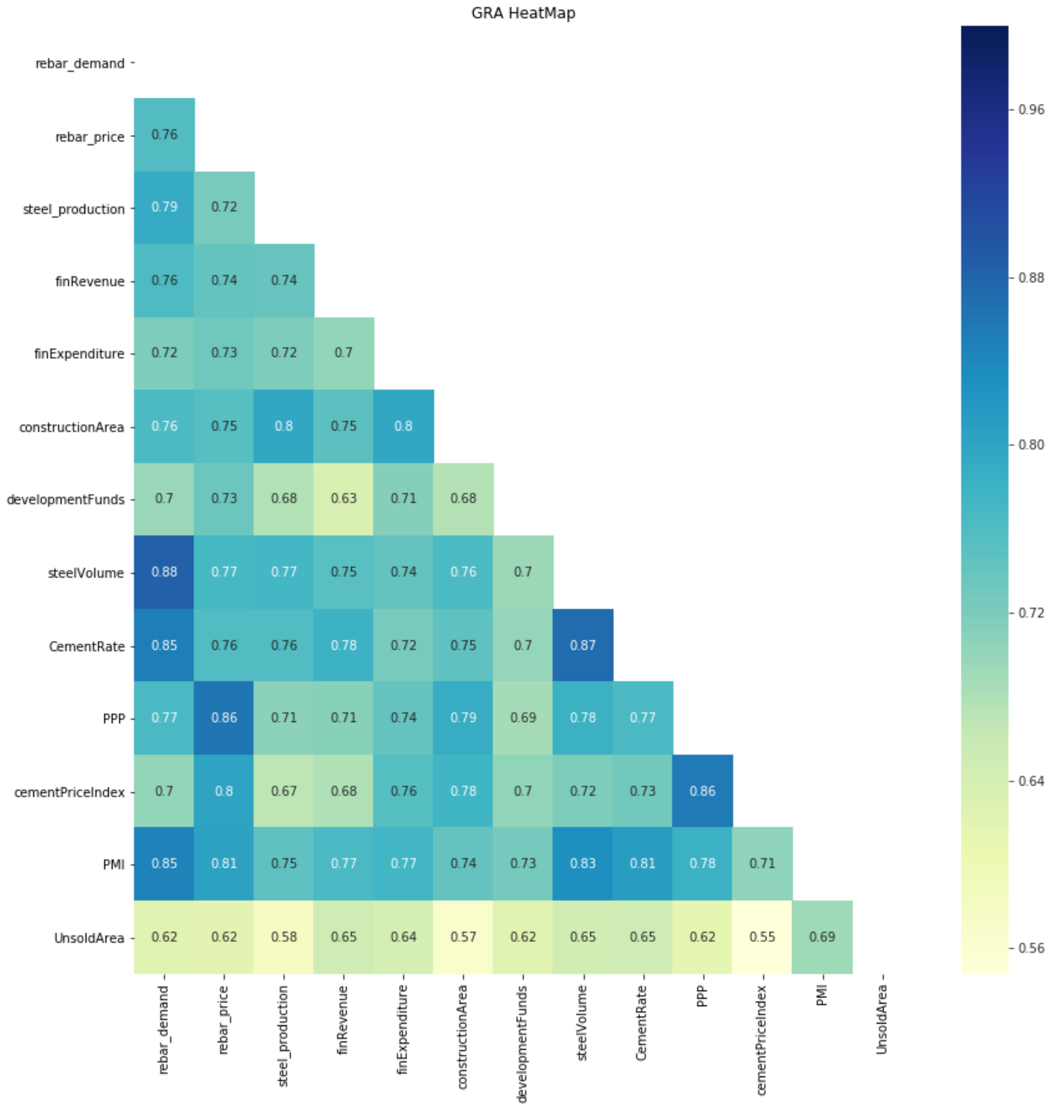


Figure 4-1 Gray correlation coefficient matrix heat map

As can be seen from Figure 4-1, the darker the color area indicates the higher the correlation between the two variables, and vice versa. We can intuitively see from the figure that among the 12 influencing factors, the demand for rebar has a strong correlation with the national construction steel transaction volume (0.88), and it is related to the cement operating rate, the construction industry PMI, and the direct supply of steel mills. , PPP project investment amount, rebar price, public financial revenue, and housing construction area also have a strong correlation (0.85, 0.85, 0.79, 0.77, 0.76, 0.76, 0.76). In other words, the national construction steel transaction volume can directly affect the demand for rebar. Cement operating rate, construction industry PMI, steel mill direct supply, PPP project investment, rebar price, public financial revenue, and housing construction area can also be affected. To a certain extent, it affects the demand for rebar, but there is no significant correlation between public financial expenditures, real estate development funding sources, cement price index, the area of commercial housing for sale and the demand for rebar (<0.76).

4.2 Model of question two

4.2.1 Model Preparation

According to the meaning of the question, first of all, according to the meaning of the question, we filter the target sub-category as the data of the rebar demand time series from March 6, 2016 to September 6, 2020, and from March 6, 2016 to 2020. The data on September 6 was clarified and preprocessed, and the time series were resampled to obtain weekly data, and the time series of rebar demand as shown in Table 1 was obtained.

Table1

Date	demand
2016/3/6	2094.926
2016/3/13	326.78592
……	……
2020/8/30	361.40449
2020/9/6	376.95

Considering that the data series we get is a typical time series and the data needs to be forecasted, we consider suggesting a differential integrated moving average autoregressive (ARIMA) model.

According to the meaning of the question, we found that many indicators such as Table 2 are provided for us to perform forecasting and analysis. Only using the ARIMA model will waste the provided data. All we establish a gray neural network model for these correlation variables, and finally adopt a combined forecast Establish an ARIMA-grey neural network model.

Table2

Date	rebar_demand	……	PMI	UnsoldArea
2016/3/6	2094.926	……	55.47097	73890.839
2016/3/13	326.78592	……	55.47097	73890.839
……	……	……	……	……
2020/8/30	361.40449	……	60.23871	50134.452
2020/9/6	376.95	……	60.2	50052

4.2.2 ARIMA Model

For non-stationary time series, after eliminating its local level or trend, it shows a certain degree of homogeneity, that is to say, some parts of the series are very similar to other parts at this time. This kind of non-stationary time series can be converted into a stationary time series after difference processing. Such a time series is called a homogeneous non-stationary time series, where the number of differences is the order of the homogeneous order.

Let ∇ be a difference operator :

$$\nabla_{yt}^{2} = \nabla(y_t - y_{t-1}) = y_t - 2y_{t-1} + y_{t-2} \quad (1)$$

For delay operator B:

$$y_{t-p} = B^p y_t, \forall_p \geq 1 \quad (2)$$

$$\nabla^k = (1-B)^k \quad (3)$$

Let y^t be a homogeneous nonstationary time series of order d, so, $\nabla^d y_t$ is a stationary time series, so we can set it as $ARIMA(p, d, q)$ model, there is

$$\lambda(B)(\nabla \, \mathrm{d}y_t) = \theta(B)\varepsilon_t \quad (4)$$

$$\lambda(B) = 1 - \lambda_1 B - \lambda_2 B^2 - \cdots - \lambda_p B^p \quad (5)$$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_p B^p \quad (6)$$

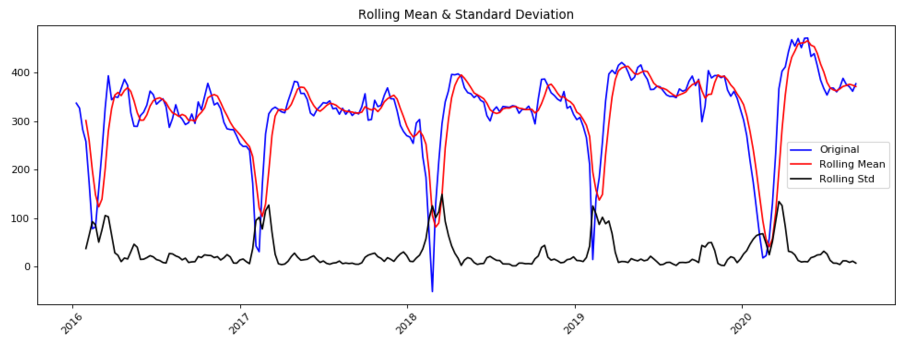
(5) and (6) are the autoregressive coefficient polynomial and the moving average coefficient polynomial, respectively. ε_t is the zero mean white noise sequence. Therefore, the $ARIMA(p, d, q)$ model can be expressed as:

$$\left(1 - \sum_{i=1}^{p} \emptyset_i L^i\right)(1-L)^d X_t = \left(1 + \sum_{i=1}^{p} \emptyset_i L^i\right)\varepsilon_t \quad (7)$$

L is the lag operator, and d is a positive integer.

1.Data Stability Test

We use Python to read the data in Table 1 to judge the stability of time series data. Make the original data and mean and variance graphs (the sliding window takes $window$=12), as shown in Picture 5-8.



Picture5-8 original data stability test chart

It can be seen intuitively from Picture 5-8 that the mean and variance of time series data show a fluctuating trend over time, that is, the original time series data is unstable, but stationarity is a prerequisite for time series analysis, so we Need to process the non-stationary series to convert it into a stationary series.

We use the ADF test method. ADF is a commonly used unit root test method. His original hypothesis is that the sequence has unit roots, that is, non-stationary. For a stable time series data, it needs to be significant at a given confidence level. Null hypothesis.

Test Statistic	-5.61862
p-value	1.16e-06
#Lags Used	1

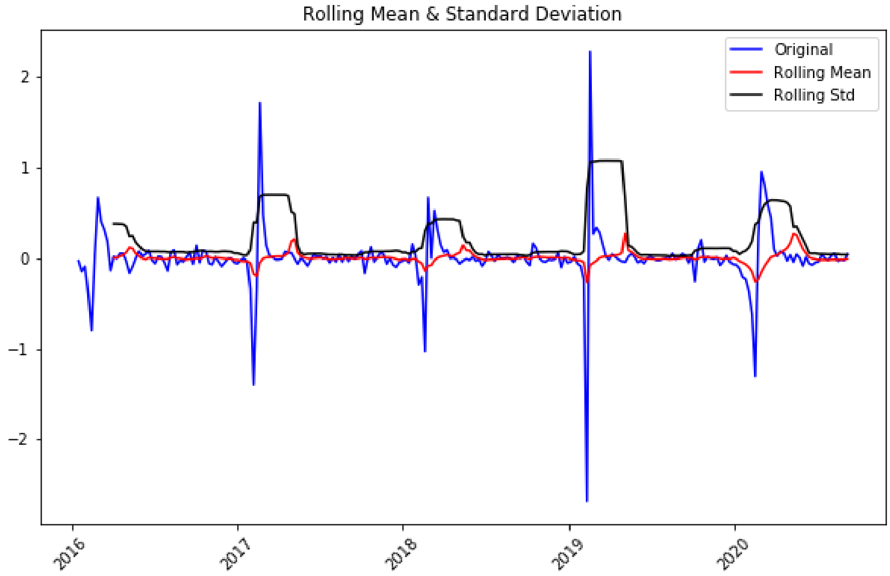
Number of Observations Used	242
Critical Value (1%)	-3.45766
Critical Value (5%)	-2.87356
Critical Value (10%)	-2.57317

2. Smoothing of time series data

We use the difference method to smooth time series.Including first difference and second difference. First-order difference and is the difference between two consecutive adjacent two items in the discrete function(Picture 5-9).

Picture 5-9 First-order difference and Second-order difference

We use Python to perform 12-order difference and then first-order difference on time series data, and Visualize the processed data distribution. The result is shown in Picture 5-10.



Picture 5-10 Data distribution diagram after difference

The variance and mean of time series data basically do not change with time, that is

In other words, the data obtained after the difference is basically stable. In order to verify the stability of the data at a deeper level,We use the unit root test Dickey-Fuller Test to make judgments. Code written in Python, the inspection The test results are shown in Table 5-11.

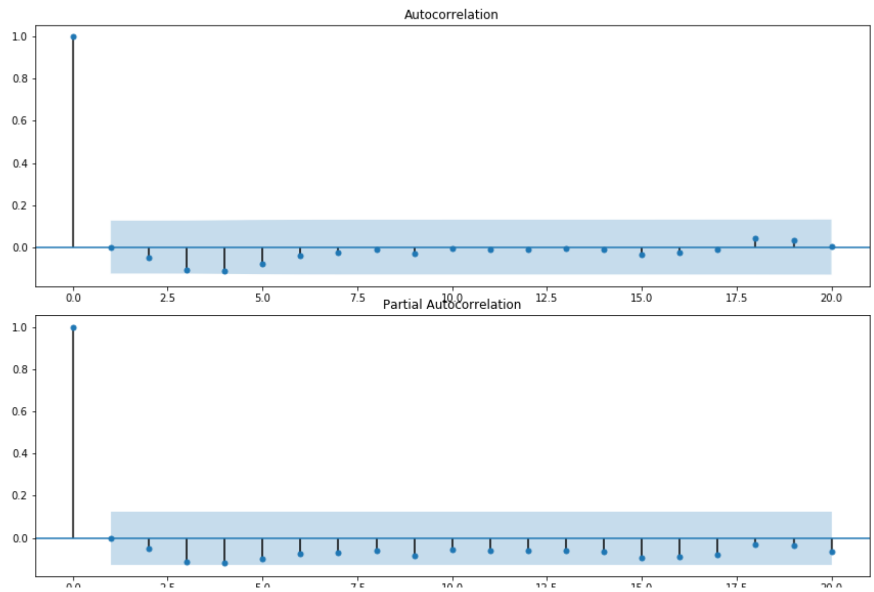
Table 5-11 Dickey-Fuller Test verifies the data after the difference

Test Statistic	-1.553287e+01
p-value	2.209014e-28
#Lags Used	0.000000e+00
Number of Observations Used	2.420000e+02
Critical Value (1%)	-3.457664e+00
Critical Value (5%)	-2.873559e+00
Critical Value (10%)	-2.573175e+00

It can be seen from Table 5-12 that, at a given test level, the value of p is much less than 0.05, indicating that the null hypothesis of data instability can be rejected. That is to say, we can conclude that after the 12th order difference, the first order The data after the difference is a smooth time series data. The mean and variance of the data tend to be constant, with almost no fluctuations (it looks steeper than before, but note that its range is only between [-0.4, 0.6]), so it can be intuitively regarded as stable data, Statistic value It is much smaller than the critical value at 1%, so the data is stable with 99% confidence.

3.Model order

After getting the stable time series data, we need to calibrate the ARIMA(p, d, q) model, that is, determine the values of p and q. We draw the autocorrelation coefficient ACF and partial autocorrelation coefficient PACF diagram of the stationary time series, The result is shown in Picture 5-12.



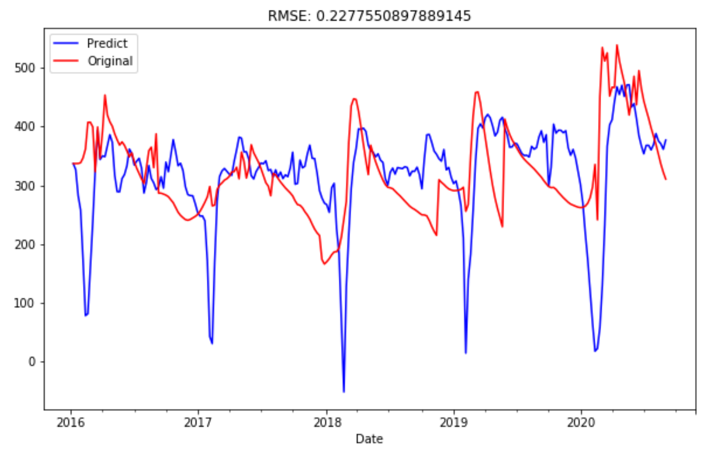
Picture 5-12 ACF and PACF diagrams for stationary timing diagrams

According to Picture 5-12, the shading in the figure represents the confidence interval, and the changes in the autocorrelation of different orders can be seen, so that the p value and q value can be selected .p is equal to 3,q is equal to 1. we use the first-order difference method, so d is equal to 1.

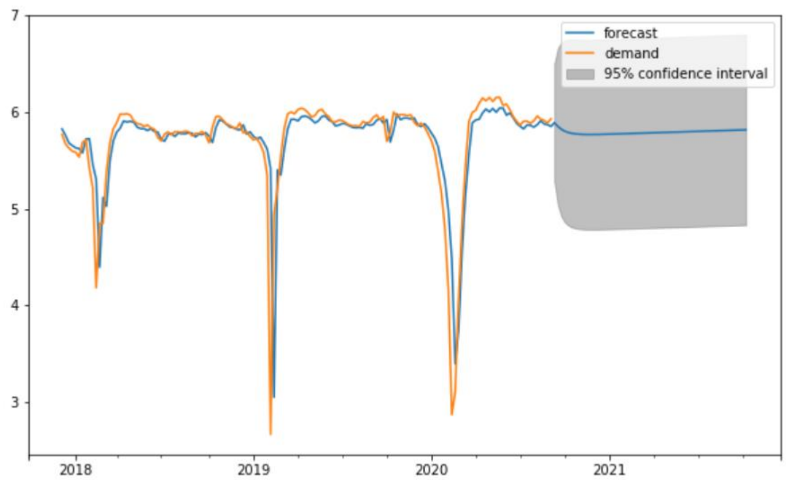
4.Model Test

After determining the order of the model, we can fit and predict the data. Since ARfits the data after relevant preprocessing, its predicted value needs to be restored by the relevant inverse transformation. We use first-order difference reduction and then perform moving average (take the sliding window wi=11) and logarithmic reduction.

Below is an image of the original data and predicted values,Picture 5-13， 5-14



Picture 5-13 Original data and predicted values



Picture 5-14 Future predicted value and confidence interval

According to the image, it can be seen that the large value of the ARIMA model can predict the result we want, but according to the RMSE value in Figure 5-13, it can be seen that the accuracy is not good enough, so we choose the combined prediction. We found that a lot of indicators were taken in the title, and the indicators with high correlation were calculated for question 1, so below we will establish a grey neural network model and finally make a combination forecast.

4.2.3 Grey Neural Network

Grey system theory is a new method to study the problems of lack of data, poor information, and uncertainty. It uses "small samples" and "poor information" uncertain systems where part of the information is known and part of the information is known as the research object. The generation and development of "partial" known information, extract valuable information, and realize the correct description and effective monitoring of system operation behavior and evolution law.

Gray neural network is a kind of combination forecasting, which belongs to series combination. Firstly, the gray model is established, and its processing ability of small samples is used to obtain the gray prediction. Then the BP neural network model is established to give play to its freedom of nonlinear and irregular data. Adapt to the learning ability, combine the output results of the gray model with the original data as the input samples of the neural network model to obtain the final combined prediction, and give full play to the advantages of the two models.

4.2.3.1GM（1,1）

Data inspection and processing

Let the reference data be listed as $x^{(0)}=(x^{(0)}(1),x^{(0)}(2),...,x^{(0)}(5))$, Calculate the grade ratio of the sequence as

$$\sigma(j)=\frac{x^{(0)}(j-1)}{x^{(0)}(j)}(j=2,3,4,5)$$

Make necessary changes to the sequence $x^{(0)}$, Make the level ratio fall within the acceptable coverage, so that the sequence meets the level ratio requirement.

Build Model $GM(1,1)$

Accumulate (ACG) the reference data column once to generate a new data column

$$x^{(1)}=(x^{(1)}(1),x^{(1)}(1)+x^{(0)}(2),...,x^{(1)}(4)+x^{(0)}(5)),$$

$$x^{(1)}(k)=\sum_{i=1}^{k}x^{(0)}(i)(k=1,2,...,5)$$

Calculate the mean series:

$$z^{(1)}(k)=0.5x^{(1)}(k)+0.5x^{(1)}(k-1),k=2,3,4,5,$$

which is $z^{(1)}=(z^{(1)}(2),z^{(1)}(3),z^{(1)}(4),z^{(1)}(5))$,

So the whitening differential equation of the model is

$$\frac{dx^{(1)}}{dt}+ax^{(1)}(t)=b$$

Record $\mu=(a,b)^T$, $Y_1=(x_i^{(0)}(2),x_i^{(0)}(3),x_i^{(0)}(4),x_i^{(0)}(5))^T$, $B=\begin{bmatrix}-z^{(1)}(2) & 1\\ -z^{(1)}(3) & 1\\ -z^{(1)}(4) & 1\\ -z^{(1)}(5) & 1\end{bmatrix}$, It is obtained by the least square method

$$\hat{\mu}=(\hat{a},\hat{b})=[B^T\cdot B]^{-1}\cdot B^T\cdot Y_1,$$

So, solving the differential equations to get

$$\hat{x}^{(1)}(k)=(x^{(1)}(1)-\frac{b}{a})\cdot e^{-ak}+\frac{b}{a}(k=2,3,4,5),$$

Then we can get the predicted value

$$\hat{x}_i^{(0)}(k)=\hat{x}_i^{(1)}(k)-\hat{x}_i^{(1)}(k-1)=(x_i^{(0)}(1)-\frac{b}{a})\cdot(e^{-ak}-e^{-a(k-1)})$$

We use the prediction result obtained by $GM(1,1)$ as the input of the BP neural network

4.2.3.2.BP

Model Theory

Theoretically, it has been proved that a network of specific deviations and at least one S-shaped hidden layer plus a linear output layer can approximate any rational function. Aiming at the measured target, the neural network model uses a three-layer feedforward network, which is composed of an input layer, a hidden layer, and an output layer. The input layer has 6 neurons, and each neuron corresponds to an influencing factor. Its input is the normalized result value of the factor; the output layer has a neuron whose output is the value of the rebar demand.

Data inspection and processing

The data must be processed with uniform dimensions and normalization before it can be used for scientific research and modeling. The data we use includes industrial chain position order, scoring value, specific quantity, etc., which need to be processed in different ways.

For the specific values and scoring of the same factor, the "unified standard range" method is used. First determine the maximum and minimum values of the sample data, and get the range L, Then determine the share Q_n of the specific factor value X, so as to unify the domain change range between 0 and 1.

The specific formula is as follows:

$$L=X_{max}-X_{min}$$

$$Q_n=(X-X_{min})/L$$

Finally, according to the normalized results of different conditions that affect and restrict the same factor, the "mean filtering" method in engineering is used to synthesize a final result. The details are as follows:

$$Q_{final}=(Q_1+Q_2+Q_3+...+Q_n)/n$$

According to the above normalization method, the corresponding digitization of the mathematical model is performed.

Bulid Model

The first layer (input layer): the input is introduced into the neural network.

$$\mathrm{Out}_i^{(1)}=\mathrm{In}_i^{(1)}=x, i=1,2,...,m$$

The second layer (hidden layer):

$$In_i^{(2)}=\sum_{i=1}^{n} w_{ij}^{(1)} \bullet Out_i^{(1)}, j=1,2,...,l$$

$$Out_j^{(2)}=f(In_j^{(2)}), j=1,2,...,l$$

Among them is the transfer function, the tangent function Sigmoid function is used here

$$f(x)=\tanh(x)$$

The third layer (output layer):

$$y_k=Out_k^{(3)}=In_k^{(3)}=\sum_{j=1}^{l} w_{jk}^{(2)} \bullet Out_j^{(2)}, k=1,2,...,n$$

Among them, m,n,l respectively represent the number of input layer nodes, the number of output layer nodes, According to this question, m=8 and n=1 can be directly determined. As for l ,it cannot be determined directly, but the empirical formula is used here:

$$l=\sqrt{m+n}+\alpha, \alpha \in [1,10]$$

To determine its initial value l=10

Solve model

For the demand model based on the BP neural network established above, we use the convenient and practical MATLAB neural network toolbox to solve it.

Data Classification:

Training	70%	166 samples
Validation	15%	35 samples
Testing	15%	35 samples

The structure diagram of BP neural network is shown in Figure 1.

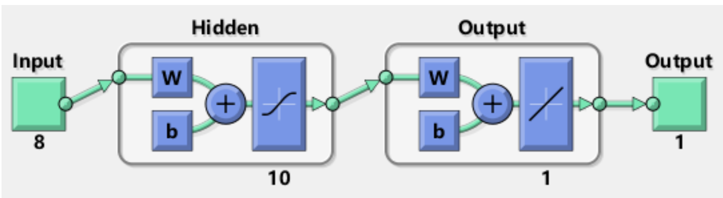


Figure 1 Neural network structure diagram

The training results are shown in Figure 2.

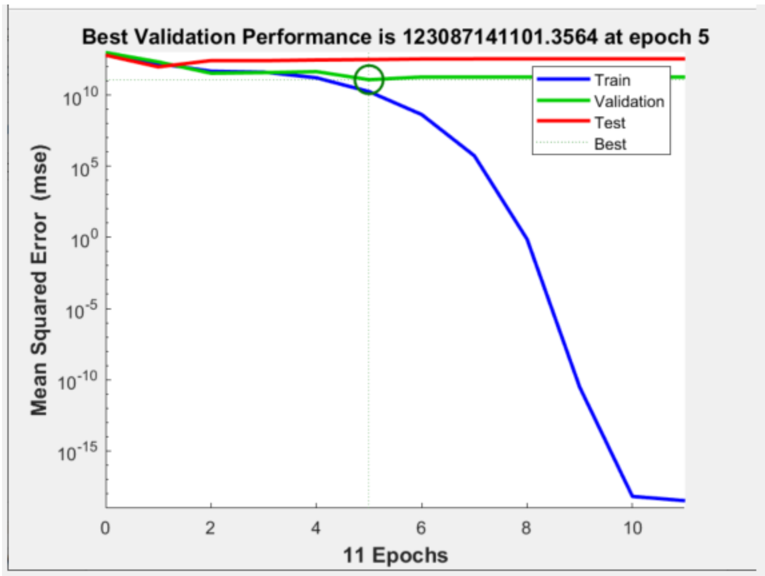


Figure 2 Training result graph

It can be seen from Figure 2 that the algorithm has reached 10^-5 through 8 trainings of mse, and 11 times it has reached a lower mse of 10^-19.

The test and verification results are shown in Figure 3

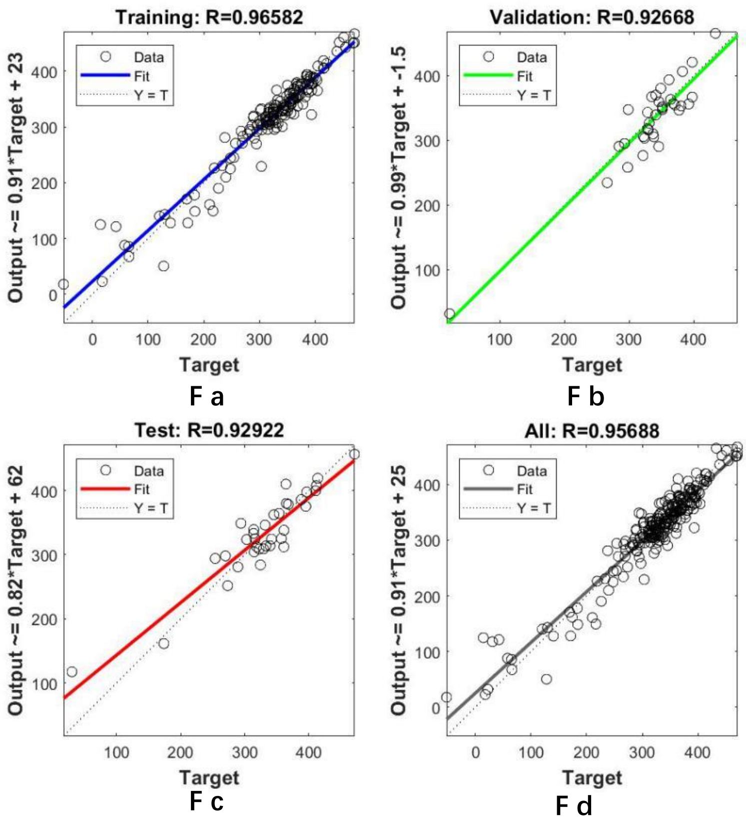


Figure a shows the fit between the training data and the Output data; Figure b shows the fit between the verification data and the Output data; Figure c shows the fit between the test data and the Output data; Figure d shows the fit between the total data and the

Out data field Degree situation

Figure 3 Test and verification results

It can be seen from Figure 3 that our BP neural network model has better realized the function of China's rebar demand forecasting, and the total fit R=0.95688.

4.2.4ARIMA- Grey neural network Model

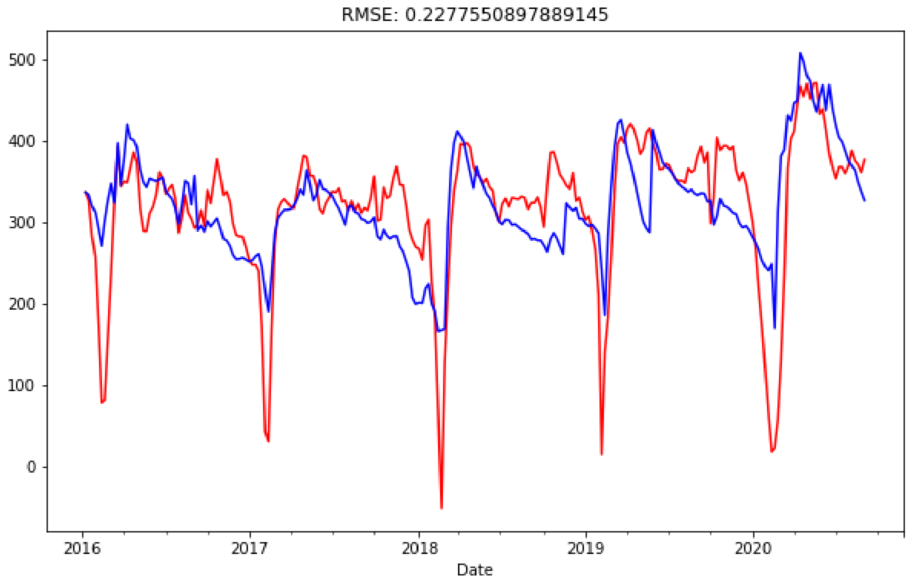
The key to the key construction process of the combined forecasting problem is to determine the weight of the single inch model used. After the construction of the ARIMA time series model and the gray neural network model is completed, the error minimization is used as the goal, the final model weight is determined, and the construction of the combined forecasting model is completed. The average absolute percentage error (MAPE) is selected as the weighted average of the inverse error The error index in the law. The average absolute percentage error of a single prediction model is as

	ARIMA	**Grey neural network**
MAPE	0.502658105668319	0.357962103933918

According to the weighted average method of reciprocal error, we get their weights respectively,

$$3.57{:}\,5.03 \approx 1{:}\,1.40$$

Get the following result graph：



5. Strengths and Weakness

5.1 Strengths

For question one: when choosing the demand of rebar indexes, we use the gray correlation analysis to verify the correlation between indicators and rebar needs through hot trying to eliminate the influence of the poor correlation between variables, so that the selected index is more normative and accuracy, more in line with the actual operation situation, is conducive to the establishment of the model.
For question two: by ARIMA time series model and grey neural network model to predict the demand for rebar and analysis, the theory of ARIMA time series model is complete, has good interpretability, gray neural network model is combined with the grey forecasting model and neural network model, this model has high forecast accuracy of the data.
For question three: By correcting the data so that the data is the real data of data update time, the model is optimized and the prediction is made again, which can make the prediction result more consistent with the actual operation situation.

5.2 Weakness

For question one： The data we initially filtered may not be the best data or may have

missed some of the more important data. And we can make use of the data is limited, to some extent is not entirely an accurate reflection of the demand for rebar, in addition to this, we obtain some data is missing, we use mathematical method to fill, but there were some error with the actual situation, so we can only use the model to more accurately predict the demand of the rebar.
For question two: we choose ARIMA time series model for data has the certain requirement, need us to do a lot of data preprocessing, which to some extent, increased the complexity and difficulty of forecasting, spent more time, gray neural network model of black box modeling process makes it interpretability of the poor, and we just use our chosen index, did not take advantage of all the indicators, however, in fact, there are a lot of the demand of rebar indicators. In addition, practical factors will also have an impact on the demand for rebar. For example, the outbreak of the epidemic this year will inevitably have an impact on the demand for rebar to a certain extent. We ignored these factors in the process of building the model, but these factors actually exist.
For question 3: We used mathematical methods to correct the data so that the data could be updated to the real data of the time. This could only make the data closer to the real data, but it was not the real data. There was still a certain degree of error, which would also lead to some errors in our prediction results.

6. Conclusion

Recommendation

With the rapid development of economy, all aspects of Our country are rapidly improving, including the increase in the demand for steel, and rebar is one of the most important steel.
Through the selection, analysis and prediction of various indicators affecting Rebar Steel from 2016 to 2020, we found that the eight indicators with the greatest impact on

Rebar Price, Steel Production, Fin Revenue, Construction Area, Steel Volume, Cement Rate, PPP and PMI respectively. In addition, it is concluded that the demand for rebar shows a relatively stable growth trend in the future. Therefore, our suggestion for the steel market is to timely adjust the price of steel and appropriately increase the production of steel. The government's income is an external uncontrollable factor. If the government's income is high, it can increase the subsidy for the steel market. If the government's income is lower than before, it will correspondingly reduce the subsidy for the steel market, so the expansion mode of steel production can be appropriately reduced.

Market supply of rebar industry refers to a certain amount of goods or services that producers are willing and able to provide at each price level in a certain period of time. Market demand of rebar industry refers to the desire of the downstream to be able to buy and willing to buy a specific commodity. It shows the quantity of a certain commodity that an individual is willing to buy in a certain period of time as the price rises and falls with other factors remaining unchanged.

But in fact, because this year appeared COVID - 19 outbreak, during the outbreak of all basic construction to cease, and during that time, the demand for steel is decreased, but after the outbreak to return to work and production, the demand for steel will greatly enhance, so from the perspective of national macroeconomic regulation and control should be leaning to steel production, we will deepen reform of the structural steel industry supply side and prevent shortage situation from the point of view of commodities trading, can be appropriate to increase investment of rebar futures. The epidemic has also had an impact on minerals and finance, which will adversely affect the steel market to a certain extent. Therefore, the price of rebar should be controlled within a certain range and should not increase or decrease too much.

In addition, due to the impact of the Trade war between China and the United States and the global market downturn, China's steel export volume has decreased, which has also had a significant impact on the production of steel market, and experts predict that the demand for rebar will also be slightly reduced.

However, the steel market will still develop in a relatively stable form, just as we predicted in the model, with a relatively stable growth trend. Although there will be some unexpected factors leading to large fluctuations, it will recover to a stable state within a certain period of time.

To sum up, proper regulation of the production and pricing of rebar is a very important means to cope with the changing demands of rebar market. After the comprehensive influence of all aspects, we come to the conclusion that the production of rebar should be expanded appropriately and the price of rebar should be raised to a small extent, but the income of enterprises in the steel market will decrease to a certain extent.

References

[1] CHAI Zhi-jun. An improved grey BP neural network prediction model.2020(10).
[2] MA Xiong-wei. Application of Grey neural network model in pork Price Forecasting. 2008(8).
[3] YANG Guo-hua. Improvement and application of GM_1_1 grey forecasting

model.2020(10).
[4] XU Jing-na. Research on tourism Demand forecast of Shanghai based on BP neural network.2020(7).
[5] YAN-Hong. Research on tourism Demand forecast of Shanghai based on BP neural network. 2020(10).
[6] DAI Xiao-ling, TANG Ming-dong, LV Sai-xia. Web service selection based on Grey correlation analysis.No.254(02):107-114. 2016.
[7] ZHANG Lan-ying, YANG Wei. Steel market review in 2019 and outlook for 2020. 2020(1).
[8] ZHANG Long-qiang, YANG Mei-mei, YU Zhi-min. Analysis and response Suggestions on the impact of COVID-19 on China's steel market. 2020(2).
[9] HEN Ke-xin. The demand for steel will explode after the epidemic. 2020(2).

Appendix

----------------Grey relational analysis model code（Python3.7）------------------

```
import pandas as pd
import numpy as np
from numpy import *
import matplotlib.pyplot as plt
%matplotlib inline


rebar = pd.read_csv("../ARIMA/rebar.csv")
rebar.head()
rebar.drop(['Date'],axis=1,inplace=True)
rebar=rebar.fillna(method='ffill')

def dimensionlessProcessing(df):
    newDataFrame = pd.DataFrame(index=df.index)
    columns = df.columns.tolist()
    for c in columns:
        d = df[c]
        MAX = d.max()
        MIN = d.min()
        MEAN = d.mean()
        newDataFrame[c] = ((d - MEAN) / (MAX - MIN)).tolist()
    return newDataFrame

def GRA_ONE(gray, m=0):

    gray = dimensionlessProcessing(gray)

    std = gray.iloc[:, m]
    gray.drop(str(m),axis=1,inplace=True)
```

```
    ce = gray.iloc[:, 0:]
    shape_n, shape_m = ce.shape[0], ce.shape[1]


    a = zeros([shape_m, shape_n])
    for i in range(shape_m):
        for j in range(shape_n):
            a[i, j] = abs(ce.iloc[j, i] - std[j])


    c, d = amax(a), amin(a)


    result = zeros([shape_m, shape_n])
    for i in range(shape_m):
        for j in range(shape_n):
            result[i, j] = (d + 0.5 * c) / (a[i, j] + 0.5 * c)


    result_list = [mean(result[i, :]) for i in range(shape_m)]
    result_list.insert(m,1)
    return pd.DataFrame(result_list)


def GRA(DataFrame):
    df = DataFrame.copy()
    list_columns = [
        str(s) for s in range(len(df.columns)) if s not in [None]
    ]
    df_local = pd.DataFrame(columns=list_columns)
    df.columns=list_columns
    for i in range(len(df.columns)):
        df_local.iloc[:, i] = GRA_ONE(df, m=i)[0]
    return df_local
data_rebar_gra = GRA(rebar)
# data_wine_gra.to_csv(path+"GRA.csv")
data_rebar_gra
data_rebar_gra.columns = rebar.columns
data_rebar_gra.index = rebar.columns
data_rebar_gra

import seaborn as sns

def ShowGRAHeatMap(DataFrame):
```

```
    colormap = plt.cm.RdBu
    ylabels = DataFrame.columns.values.tolist()
    f, ax = plt.subplots(figsize=(14, 14))
    ax.set_title('GRA HeatMap')

    mask = np.zeros_like(DataFrame)
    mask[np.triu_indices_from(mask)] = True

    with sns.axes_style("white"):
        sns.heatmap(DataFrame,
                    cmap="YlGnBu",
                    annot=True,
                    mask=mask,
                   )
    plt.savefig('GRA.png')

    plt.show()
ShowGRAHeatMap(data_rebar_gra)
```

```
----------------ARIMA（python3.7 Anaconda3）----------------------------
from datetime import datetime
import numpy as np                #for numerical computations like log,exp,sqrt etc
import pandas as pd               #for reading & storing data, pre-processing
import matplotlib.pylab as plt #for visualization
#for making sure matplotlib plots are generated in Jupyter notebook itself
%matplotlib inline
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.stattools import acf, pacf

from statsmodels.tsa.seasonal import seasonal_decompose

from statsmodels.tsa.arima_model import ARIMA
from statsmodels.tsa.arima_model import ARMA
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 10, 6
Data = pd.read_csv('demand-data.csv', index_col=0)

Data = Data.reset_index()

#Parse strings to datetime type
#convert from string to datetime ###Name: date, dtype: datetime64[ns]
Data['Date']= pd.to_datetime(Data['Date'],infer_datetime_format=True)
```

```
indexedDataset = Data.set_index(['Date'])
indexedDataset = indexedDataset.iloc[::]
indexedDataset.drop(['index'],axis=1,inplace=True)
indexedDataset.drop(indexedDataset.head(3).index,inplace=True)
indexedDataset.drop(indexedDataset.tail(61).index,inplace=True)
indexedDataset
PA=pd.read_csv('predictions_ARIMA1.csv', index_col=0)
R=indexedDataset.resample('D').interpolate()
R=R.resample('W').mean()
print(R)
# plot graph
plt.figure(figsize=(15,5),dpi = 80)
plt.xticks(rotation=45)
plt.xlabel('Date')
plt.ylabel('demand')
plt.plot(R)

indexedDataset = indexedDataset.head(200)
rolmean = R.rolling(window=4).mean()
rolstd = R.rolling(window=4).std()
#print(rolmean,rolstd)
rolmean
#Plot rolling statistics


plt.figure(figsize=(15,5),dpi = 80)
plt.xticks(rotation=45)
orig = plt.plot(R.ix[:,0], color='blue', label='Original')
mean = plt.plot(rolmean, color='red', label='Rolling Mean')
std = plt.plot(rolstd, color='black', label='Rolling Std')

plt.legend(loc='best')
plt.title('Rolling Mean & Standard Deviation')
plt.savefig('1.png')
plt.show(block=False)

def test_stationarity(timeseries):

    #Determine rolling statistics
    movingAverage = timeseries.rolling(window=12).mean()
    movingSTD = timeseries.rolling(window=12).std()

    #Plot rolling statistics
    orig = plt.plot(timeseries, color='blue', label='Original')
```

```
    mean = plt.plot(movingAverage, color='red', label='Rolling Mean')
    std = plt.plot(movingSTD, color='black', label='Rolling Std')
    plt.xticks(rotation=45)
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.savefig('t.png')
    plt.show(block=False)

    #Perform Dickey–Fuller test:    Augmented Dickey–Fuller ADF
    print('Results of Dickey Fuller Test:')
    dftest = adfuller(timeseries['demand'], autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])
    for key,value in dftest[4].items():
        dfoutput['Critical Value (%s)'%key] = value
    print(dfoutput)

###3.2    Differencing -Timeshift transformation
#Estimating trend
indexedDataset_logScale = np.log(R)    #taking log
plt.plot(indexedDataset_logScale)

##Differencing

datasetLogDiffShifting_1 = indexedDataset_logScale - indexedDataset_logScale.shift()
plt.subplot(2,1,1)
plt.plot(datasetLogDiffShifting_1,color='y')
plt.legend(' first differencefirst difference')


datasetLogDiffShifting_2 = datasetLogDiffShifting_1 - datasetLogDiffShifting_1.shift()
plt.subplot(2,1,2)
plt.plot(datasetLogDiffShifting_2,color='b')
plt.savefig('2.png')
plt.legend('second difference')
example1 = indexedDataset_logScale.diff(1)
plt.plot(example1)
example2 = example1.diff(1)
plt.plot(example2)
example3 = example2.diff(1)
plt.plot(example3)
datasetLogDiffShifting_1.dropna(inplace=True)。
test_stationarity(datasetLogDiffShifting_1)
```

```
datasetLogDiffShifting_2.dropna(inplace=True)
test_stationarity(datasetLogDiffShifting_2)

import statsmodels.api as sm
fig = plt.figure(figsize=(12,8))

#acf    from statsmodels.tsa.stattools import acf, pacf
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(datasetLogDiffShifting_1, lags=20,ax=ax1)
ax1.xaxis.set_ticks_position('bottom')
fig.tight_layout();

#pacf
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(datasetLogDiffShifting_1, lags=20, ax=ax2)
ax2.xaxis.set_ticks_position('bottom')
fig.savefig('t.png')
fig.tight_layout();

indexedDataset_logScale.fillna(method='bfill',inplace=True)
model_3 = ARIMA(indexedDataset_logScale, order=(3,1,2))
# indexedDataset_logScale.to_csv('ss.csv')
results_ARIMA = model_3.fit()
plt.plot(datasetLogDiffShifting_1)
plt.plot(results_ARIMA.fittedvalues, color='red')
# print(sum((results_ARIMA.fittedvalues - datasetLogDiffShifting_1['demand'])**2))
plt.title('RSS:                    %.4f'%sum((results_ARIMA.fittedvalues -
datasetLogDiffShifting_1['demand'])**2))



predictions_ARIMA_diff = pd.Series(results_ARIMA.fittedvalues, copy=True)
predictions_ARIMA_diff

#Convert to cumulative sum
predictions_ARIMA_diff_cumsum = predictions_ARIMA_diff.cumsum()
predictions_ARIMA_diff_cumsum =predictions_ARIMA_diff_cumsum.cumsum()
print(predictions_ARIMA_diff_cumsum)


predictions_ARIMA_log    =    pd.Series(indexedDataset_logScale['demand'].iloc[0],
index=indexedDataset_logScale.index)
predictions_ARIMA_log                                                          =
predictions_ARIMA_log.add(predictions_ARIMA_diff_cumsum, fill_value=0)
```

```
predictions_ARIMA_log.tail(40)
predictions_ARIMA = np.exp(predictions_ARIMA_log)
results_ARIMA.plot_predict(50,300)

--------------------Time series data preprocessing Code(Python3.7 Anaconda3)----------
-
from datetime import datetime
import numpy as np                    #for numerical computations like log,exp,sqrt etc
import pandas as pd                   #for reading & storing data, pre-processing
import matplotlib.pylab as plt #for visualization
#for making sure matplotlib plots are generated in Jupyter notebook itself
%matplotlib inline
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.stattools import acf, pacf

##(decomposing)
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.tsa.arima_model import ARMA
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 10, 6
Data = pd.read_csv('demand-data.csv', index_col=0)
Data = Data.reset_index()
#Parse strings to datetime type
#convert from string to datetime ###Name: date, dtype: datetime64[ns]
Data['Date']= pd.to_datetime(Data['Date'],infer_datetime_format=True)

indexedDataset = Data.set_index(['Date'])
indexedDataset = indexedDataset.iloc[::]
indexedDataset.drop(['index'],axis=1,inplace=True)
indexedDataset.drop(indexedDataset.head(3).index,inplace=True)
indexedDataset.drop(indexedDataset.tail(61).index,inplace=True)
indexedDataset.columns
ts = pd.Series(indexedDataset['demand'].values,index=indexedDataset.index)
R=ts.resample('D').interpolate()
# R=ts.resample('D').interpolate()
Rx=R.resample('W').mean()

def getMonthTimeSeries(path):
    f = open(path,encoding='UTF-8')
    finput=pd.read_csv(f)
    finput['2016'].index=pd.date_range('2016',periods=12,freq='M')
    stemp=finput['2016']
    l = finput.columns[1:-1][::-1]
```

```
    for i in l:
        finput[i].index=pd.date_range(i,periods=12,freq='M')
        stemp=pd.concat([stemp,finput[i]])
    stemp=stemp[:-4]
    stemp=stemp.resample('M').interpolate()
    stemp=stemp.resample('D').interpolate()
    stemp=stemp.resample('W').mean()
    return stemp[5:]

finRevenue=getMonthTimeSeries('./8Data/公共财政收入：当月值-季节性图表.csv')
finRevenue.name='finRevenue'
finExpenditure=getMonthTimeSeries('./8Data/公共财政支出：当月值-季节性图表.csv')
finExpenditure.name='finExpenditure'
landPurchased=getMonthTimeSeries('./8Data/本年购置土地面积：当月值-季节性图表.csv')
landPurchased.name='landPurchased'
constructionArea=getMonthTimeSeries('./8Data/房屋施工面积：当月值-季节性图表.csv')
constructionArea.name='constructionArea'
developmentFunds=getMonthTimeSeries('./8Data/房地产开发资金来源：合计：累计值-季节性图表.csv')
developmentFunds.name='developmentFunds'

def getDayTimeSeries(path):
    f = open(path,encoding='UTF-8')
    finput=pd.read_csv(f)
    t= finput.columns[1:-1]
    t=t[::-1]
    l= '2016'+"-"+finput['Category']
    pd.to_datetime(l,infer_datetime_format=True)
    finput['2016'].index=pd.to_datetime(l,infer_datetime_format=True)
    ll=finput['2016']
    for i in t:
        tt= i+"-"+finput['Category']
        finput[i].index=pd.to_datetime(tt,infer_datetime_format=True)
        ll=pd.concat([ll,finput[i]])
    ll=ll.resample('D').interpolate()
    ll=ll.resample('W').mean()
    return ll['2016-03-06':'2020-09-06']

steelVolume=getDayTimeSeries("./8Data/全国建筑钢材成交量-季节性图表.csv")
steelVolume.name='steelVolume'
CementRate=getDayTimeSeries("./8Data/水泥开工全国算数平均-季节性图表.csv")
```

```
CementRate.name='CementRate'

def get3MonthTimeSeries(path):
    f = open(path,encoding='UTF-8')
    finput=pd.read_csv(f)
    t=finput["贷款需求指数:制造业"]
    Date=finput['Category']
    l=pd.Series(t)
    l.index=pd.to_datetime(Date,infer_datetime_format=True)
    l=l.resample('D').interpolate()
    l=l.resample('W').mean()
    return l['2016-03-06':]
loanDemandIndex=get3MonthTimeSeries("./8Data/贷款需求指数.csv")
loanDemandIndex.name='loanDemandIndex'

def getPPPTimeSeries(path):
    f = open(path,encoding='UTF-8')
    finput=pd.read_csv(f)
    finput
    t=finput["PPP 项目投资额:总投资"]
    Date=finput['Category']
    l=pd.Series(t)
    l.index=pd.to_datetime(Date,infer_datetime_format=True)
    l=l.resample('D').interpolate()
    l=l.resample('W').mean()
    return l['2016-03-06':]
PPP=getPPPTimeSeries("./8Data/PPP 项目投资额.csv")
PPP.name='PPP'
def getCementpriceindexTimeSeries(path):
    finput=pd.read_excel(path)
    finput
    t=finput["水泥价格指数:全国"]
    Date=finput['Category']
    l=pd.Series(t)
    l.index=pd.to_datetime(Date,infer_datetime_format=True)
    l=l.resample('D').interpolate()
    l=l.resample('W').mean()
    # finput
    return l['2016-03-06':'2020-09-06']
cementPriceIndex=getCementpriceindexTimeSeries("./8Data/水泥价格和螺纹钢价格.xlsx")
cementPriceIndex.name='cementPriceIndex'

def getPMITimeSeries(path):
```

```
    f = open(path,encoding='UTF-8')
    finput=pd.read_csv(f)
    t=finput["非制造业 PMI:建筑业"]
    Date=finput['Category']
    l=pd.Series(t)
    l.index=pd.to_datetime(Date,infer_datetime_format=True)
    l=l.resample('D').interpolate()
    l=l.resample('W').mean()
    # finput
    return l['2016-03-06':'2020-09-06']
PMI=getPMITimeSeries("./8Data/非制造业 PMI：建筑业.csv")
PMI.name='PMI'

def getUnsoldAreaTimeSeries(path):
    f = open(path,encoding='UTF-8')
    finput=pd.read_csv(f)
    t=finput["商品房待售面积:累计值"]

    Date=finput['Category']
    l=pd.Series(t)
    l.index=pd.to_datetime(Date,infer_datetime_format=True)
    l=l.resample('D').interpolate()
    l=l.resample('W').mean()
    # finput
    return l['2016-03-06':]
UnsoldArea=getUnsoldAreaTimeSeries("./8Data/商品房待售面积：累计值.csv")
UnsoldArea.name='UnsoldArea'

def getSteelSupplyTimeSeries(path):

    f = open(path,encoding='UTF-8')
    finput=pd.read_csv(f)
    t= finput.columns[1:-1]
    t=t[::-1]
    l= '2017'+"-"+finput['Category']
    pd.to_datetime(l,infer_datetime_format=True)
    finput['2017'].index=pd.to_datetime(l,infer_datetime_format=True)
    ll=finput['2017']
    for i in t:
        tt= i+"-"+finput['Category']
        finput[i].index=pd.to_datetime(tt,infer_datetime_format=True)
        ll=pd.concat([ll,finput[i]])
    ll=ll.resample('D').interpolate()
    ll=ll.resample('W').mean()
```

```
    # ll=ll.resample('W').bfill(12)
    return ll[:"2020-9-6"]
steelSupply=getSteelSupplyTimeSeries("./8Data/钢厂直供量-季节性图片.csv")
steelSupply

a=pd.read_excel('./Data/价格和基差/螺纹钢价格和基差/螺纹钢价格.xlsx')
temp=a['Category']
temp0=pd.to_datetime(temp,infer_datetime_format=True)
temp1=a['全国均价']
rebar_price=pd.Series(temp1)
rebar_price.index=temp0
# rebar_price
# rebar_price['2016':]
# temp
rebar_price=rebar_price['2016':'2020-09-06'].resample('W').mean()
rebar_price.name='rebar_price'
rebar_price=rebar_price['2016-3-6':]


steel_production=pd.read_excel('./8Data/钢材产量.xlsx')
q=steel_production['Category']
q0=pd.to_datetime(q,infer_datetime_format=True)
q1=steel_production['钢厂直供量（万吨）']
steel_production=pd.Series(q1)
steel_production.index=q0
steel_production=steel_production.resample('D').interpolate()
steel_production=steel_production.resample('W').mean()
steel_production.name='steel_production'
steel_production=steel_production['2016-3-6':'2020-9-6']
steel_production

rebar_demand=Rx['2016-3-06':]
rebar_demand.name='rebar_demand'
gxx=pd.concat([rebar_demand,rebar_price,steel_production,finRevenue,finExpenditu
re,constructionArea,developmentFunds,steelVolume,CementRate,PPP,cementPriceInd
ex,PMI,UnsoldArea],axis=1)
gxx.to_csv('rebar.csv')
```

-------------Gm(1,1) Code MATLAB R2018a-------------------

```
syms a u;
c=[a,u]';
load('r.mat')
```

```
A=rebar(:,5)
A=A'
Q=3*log(A+8)
A=Q
Ago=cumsum(A);
n=length(A);
for k=1:(n-1)
    Z(k)=(Ago(k)+Ago(k+1))/2;
end
Yn =A;
Yn(1)=[];
Yn=Yn';
E=[-Z;ones(1,n-1)]';
c=(E'*E)\(E'*Yn);
c= c';
a=c(1)
u=c(2)
F=[];
F(1)=A(1);
for k=2:(n)
    F(k)=(A(1)-u/a)/exp(a*(k-1))+u/a;
end
G=[];
G(1)=A(1);
for k=2:(n)
    G(k)=F(k)-F(k-1);
end
t1=1:n;
t2=1:n;
plot(t1,A,'bo--');
hold on;
plot(t2,G,'r*-');
title('prd);
legend('real','prd');

e=A-G;
q=e/A;
s1=var(A);
s2=var(e);
c=s2/s1;
len=length(e);
p=0;
for i=1:len
    if(abs(e(i))<0.6745*s1)
```

```
        p=p+1;
    end
end
p=p/len;



-----------------BP Neural Network Code MATLAB R2018a--------
load("test1.mat")
hiddenLayerSize = 10;


trainFcn = 'trainlm';

net = fitnet(hiddenLayerSize,trainFcn);


net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;


[net,tr] = train(net,p,t);


tp = sim(net,p);


performance = perform(net,t,tp);


view(net)
```