1, The cost function in logistic regression, also known as the logistic loss or cross-entropy loss, measures how well the logistic regression model's predictions match the actual labels of the training data. The goal of training a logistic regression model is to find the parameters (weights) that minimize this cost function.

For a single training pair: $\hat{y} = P(Y = 1|X = x) = \sigma(w \cdot x + b)$, where $\sigma(z) = \frac{1}{1+e^z}$

$$P(y|x) = \begin{cases} \hat{y} & if\ y = 1 \\ (1 - \hat{y}) & if\ y = 0 \end{cases}$$

If the labels are 0 and 1, then Y is a Bernoulli random variable Y ~ Ber(p), where p = $\sigma(w \cdot x + b)$

so,P(y|x) = $\hat{y}^y(1 - \hat{y})^{1-y}$, Taking the log of both sides without changing their monotonicity yields:

$$logP(y|x) = ylog(\hat{y}) + (1 - y)log(1 - \hat{y}) = -L(\hat{y}, y)$$

The cost function for logistic regression is derived from the likelihood of the observed data. The likelihood is the probability of observing the given set of labels $y_1, y_2, y_3 \ldots y_n$ for the given set of inputs $x_1, x_2, x_3 \ldots x_n$ given the model parameters w and b, To derive the cost function, we sum the negative log-likelihoods over all training examples and then take the average:

$$J(w, b) = -log \prod_{i=1}^{n} p(y^{[i]}|x^{[i]}) = -\frac{1}{n} \sum_{i=1}^{n} \left(y^{[i]} log(\hat{y}^{[i]}) + (1 - y^{[i]}) log(1 - \hat{y}^{[i]}) \right)$$

Putting it all together, the cost function J(w,b) for logistic regression is:

$$J(w, b) = -\frac{1}{i} \sum_{n=1}^{n} \left(y^{[i]} log(\sigma(w^T \cdot x^{[i]} + b)) + (1 - y^{[i]}) log(1 - \sigma(w^T \cdot x^{[i]} + b)) \right)$$

where:

i is the number of training examples.

$y^{[i]}$ is the actual label for the i-th training example

$\sigma(w^T \cdot x^{[i]} + b)$ is the predicted probability for the i-th training example

when $y^{[i]} = 1$, the term $(1 - y^{[i]}) log(1 - \hat{y}^{[i]})$ drops out, and the cost function focuses on $log(\hat{y}^{[i]})$,If the model predicts a low probability for $y^{[i]} = 1$, the cost will be high, encouraging the model to increase $\hat{y}^{[i]}$

when $y^{[i]} = 0$, the term $y^{[i]} log \hat{y}^{[i]}$ drops out, and the cost function focuses on $1 - log(\hat{y}^{[i]})$,If the model predicts a low probability for $y^{[i]} = 0$, the cost will be high, encouraging the model to decrease $\hat{y}^{[i]}$

By minimizing this cost function using optimization techniques like gradient descent, the logistic regression model learns the optimal parameters w and b that best fit the training data

2 Voting classifiers are a type of ensemble learning method used to improve the performance and robustness of predictive models by combining the predictions of multiple base models (svc, LogisticRegression,DecisionTreeClassifier ...). The main idea behind voting classifiers is that by aggregating the predictions of several models, the overall prediction is more accurate and less likely to be influenced by the weaknesses of individual models, There are two main types of voting methods in ensemble learning: hard voting and soft voting

In hard voting, also known as majority voting, each base classifier makes a prediction (a class label). The final prediction of the ensemble is the class label that receives the majority of votes from the base classifiers

In soft voting, each base classifier outputs a probability for each class. The final prediction is made by averaging the predicted probabilities and selecting the class with the highest average probability. This method takes into account the confidence of each classifier's predictions

The benefits of Voting Classifiers is: Improved Accuracy,Reduced Overfitting,Robustness and Versatility; The limitations is: Complexity,Computational Cost and Interpretability

Voting classifiers are a powerful technique in ensemble learning, leveraging the strengths of multiple models to produce more accurate and reliable predictions. By combining the outputs of several models, voting classifiers can significantly improve the performance and robustness of predictive models in various applications

3,Use the seven-step method to build a model to predict the rise and fall of the Shanghai Composite Index

1, Data Collection

Download the Shanghai Composite Index trading data from 2016/1/1 to 2024/6/3 through Yahoo Finance

2, Data Preprocessing

Use the StandardScaler function to standardize the data

3,Feature Engineering

Select features: 'HC', 'Sign', 'RET', 'VMA_5', 'VMA_10', 'VMA_20', 'VMA_60', 'VMA_120', 'OC', 'OC7', 'OC14', 'HL', 'HC7', 'HC4', 'STD', 'SMA_5', 'SMA_10', 'SMA_20', 'SMA_60', 'SMA_120', 'EMA_5', 'EMA_10', 'EMA_20', 'EMA_60',

'EMA_120', 'Momentum__5', 'Momentum__10', 'Momentum__20', 'Momentum__60', 'Momentum__120'

Optimize the features and remove the features with correlation greater than 0.9 to obtain the features: 'HC', 'Sign', 'RET', 'VMA_5', 'VMA_10', 'VMA_60', 'OC', 'OC7', 'OC14', 'HL', 'HC7', 'STD', 'SMA_5', 'Momentum__5', 'Momentum__10', 'Momentum__20', 'Momentum__60', 'Momentum__120'

4,Model Selection

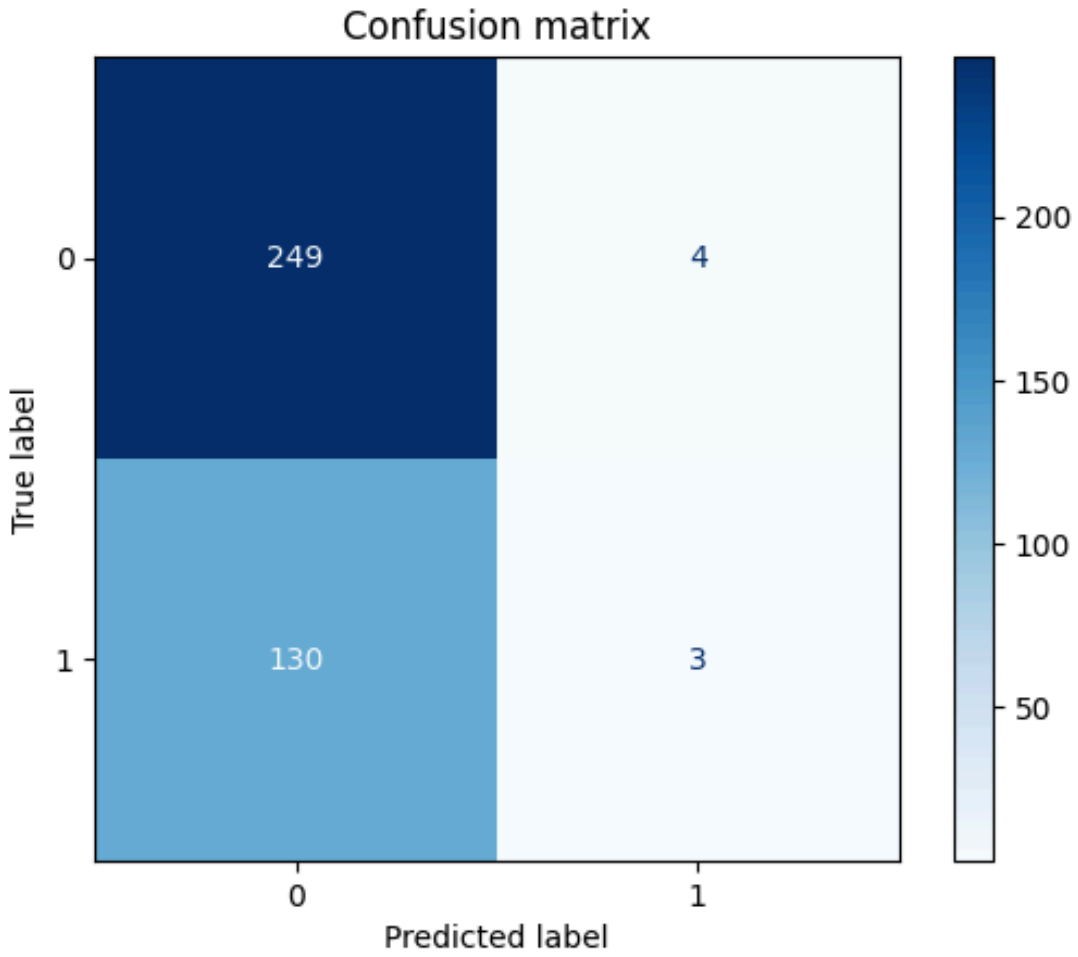
leabel uses SVM classifier to fit features

5,Model Training

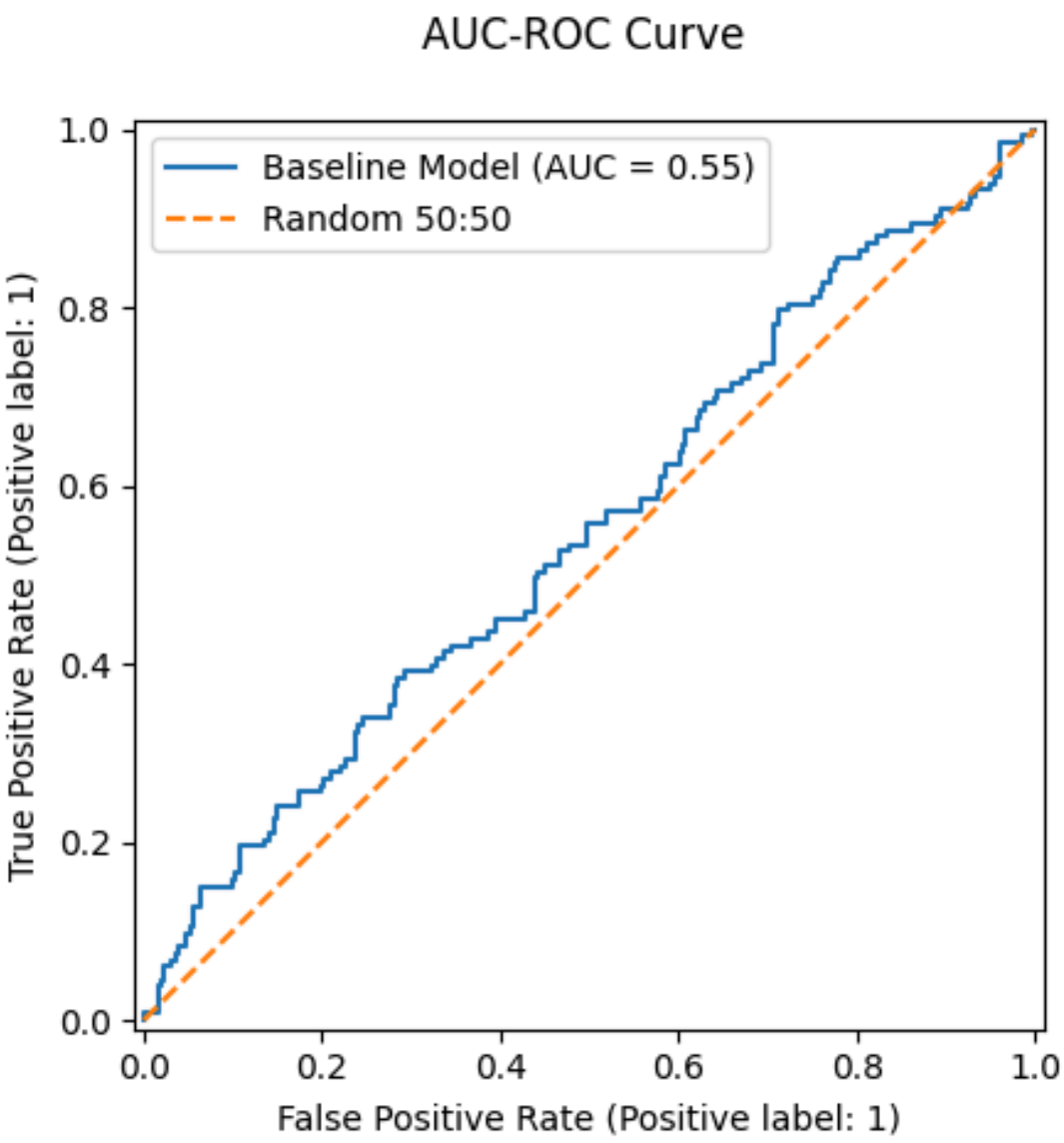
Use 80% of the data for training and 20% of the data for training

6，Model Evaluation

The results of using the svc model without parameter adjustment are Train Accuracy: 0.6708, Test Accuracy: 0.6528. It can be seen that the accuracy of both training data and test data is not very good. It can be seen from the confusion matrix that the accuracy of predicting a decline is very high when the price falls further., while the prediction of the rise is very poor. Overall, the prediction is not optimistic.



From the AUC-ROC curve, we can see that the AUC value is 0.55, which is basically random, a little higher than random guessing, and the model is not ideal.

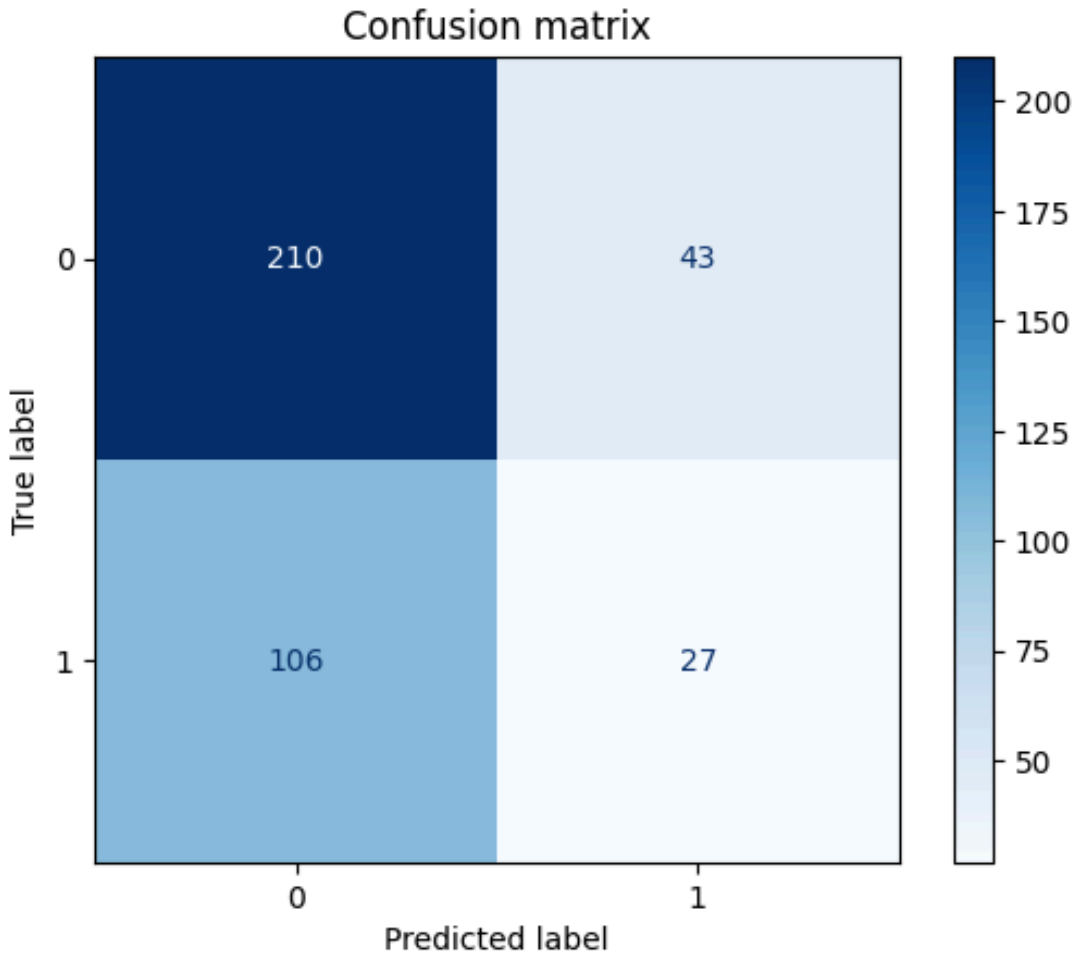


It can be seen from the report that the prediction accuracy for decline is relatively high, with a recall rate of 0.98, while the prediction accuracy for rise is very low, with a recall rate of almost 0, an average prediction accuracy probability of 0.4, and a weighted average prediction accuracy probability of 0.52. From the data, the prediction for decline is still extremely accurate, while the prediction for rise is very low, and it is almost impossible to predict a rise. This may be related to the condition of rising being higher than 0.25 of the previous day. If this is used as an investment strategy, there should be a relatively high winning rate, but in the case of long positions, there are only a few transactions in a year.

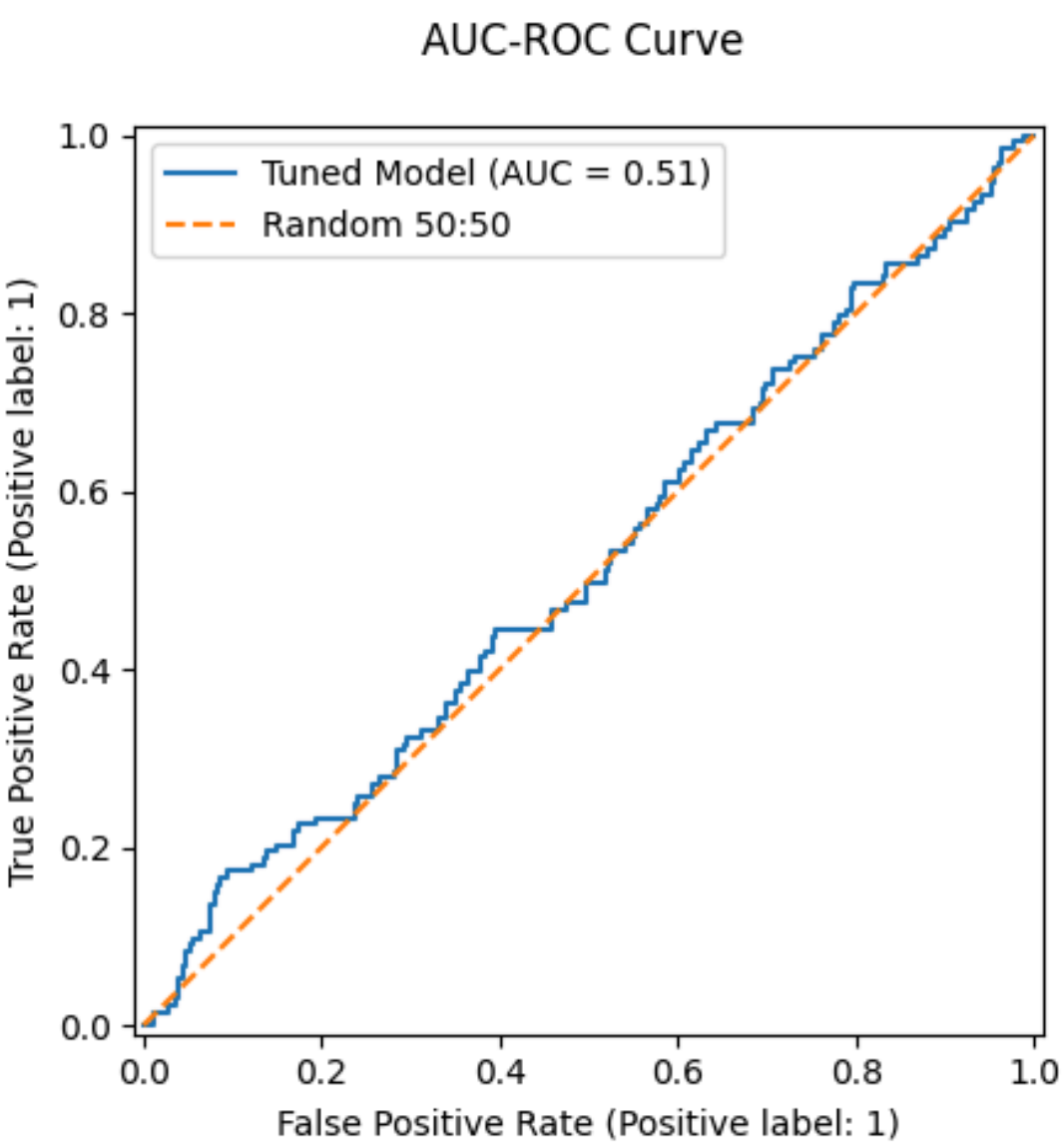
	precision	recall	f1-score	support
0	0.65	0.98	0.79	253
1	0.20	0.01	0.01	133
accuracy			0.65	386

macro avg	0.43	0.50	0.40	386
weighted avg	0.50	0.65	0.52	386

Optimize the parameters tol and C and get 'tol': 0.009878566632491585, 'C': 9.930264730821525. Use the adjusted parameters to test and get Training Accuracy: 0.8177, Test Accuracy: 0.614. It can be seen that after the parameter adjustment, the training accuracy becomes higher, while the test accuracy becomes lower. The confusion matrix after the parameter adjustment shows that the parameter adjustment also has some effect, that is, the prediction probability of the increase is a little higher, but the prediction accuracy is still very low.



From the AUC-ROC curve, we can see that the AUC value is 0.51, which is basically random, a little higher than random guessing, Adjusting the parameters did not bring better results.



After adjusting the parameters, it can be seen from the data report that the prediction accuracy for decline is relatively high, and the recall rate is only 0.7 lower than before the adjustment. The prediction accuracy for rise is increased to 0.36, the recall rate is greatly increased to 0.32, the average accuracy is increased to 0.51, and the weighted average accuracy is increased to 0.56. After the adjustment, except for the decrease in the prediction accuracy of decline, the overall prediction accuracy is improved, The effect is quite good after adjusting the parameters. but if this model is used, it is almost no different from guessing. It may have something to do with the Chinese market,We still need to find better models to adapt to the Chinese market

	precision	recall	f1-score	support
0	0.66	0.70	0.68	253
1	0.36	0.32	0.34	133
accuracy			0.57	386
macro avg	0.51	0.51	0.51	386
weighted avg	0.56	0.57	0.56	386