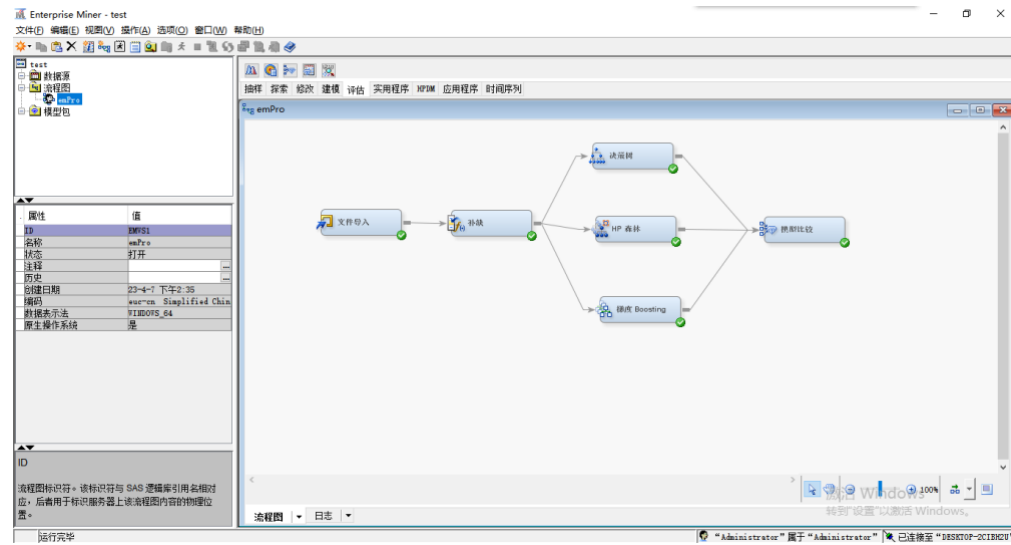


# WQD7005 DATA MINING

## SAS e-Miner:



## Data collection and import

The screenshot shows the '变量 - RIMPORT' (Variables - RIMPORT) dialog box. It contains a table with the following columns: '名称' (Name), '角色' (Role), '水平' (Level), '报表' (Report), '顺序' (Order), '删除' (Delete), '下限' (Lower Bound), and '上限' (Upper Bound). The table lists various variables and their properties.

名称	角色	水平	报表	顺序	删除	下限	上限
Age	拒绝	区间型	否		否	-	-
Churn	目标	二值型	否		否	-	-
Customer_Age	输入	区间型	否		否	-	-
Customer_ID	拒绝	区间型	否		否	-	-
Customer_Name	拒绝	列名型	否		否	-	-
Gender	输入	列名型	否		否	-	-
Payment_Method	输入	列名型	否		否	-	-
Product_Category	输入	列名型	否		否	-	-
Product_Price	输入	区间型	否		否	-	-
Purchase_Date	拒绝	区间型	否		否	-	-
Quantity	输入	区间型	否		否	-	-
Returns	输入	列名型	否		否	-	-
Total_Purchase_Amount	输入	区间型	否		否	-	-

## Data cleaning and preprocessing

The screenshot shows the '结果 - 节点: 补缺' (Results - Node: Imputation) window. It contains a table with the following columns: '变量名称' (Variable Name), '补缺方法' (Imputation Method), '补缺变量' (Imputation Variable), '补缺值' (Imputation Value), '角色' (Role), '测量水平' (Measurement Level), '标签' (Label), and 'TRAIN的缺失个数' (Number of Missing Values in TRAIN). The table shows results for the 'Returns' variable.

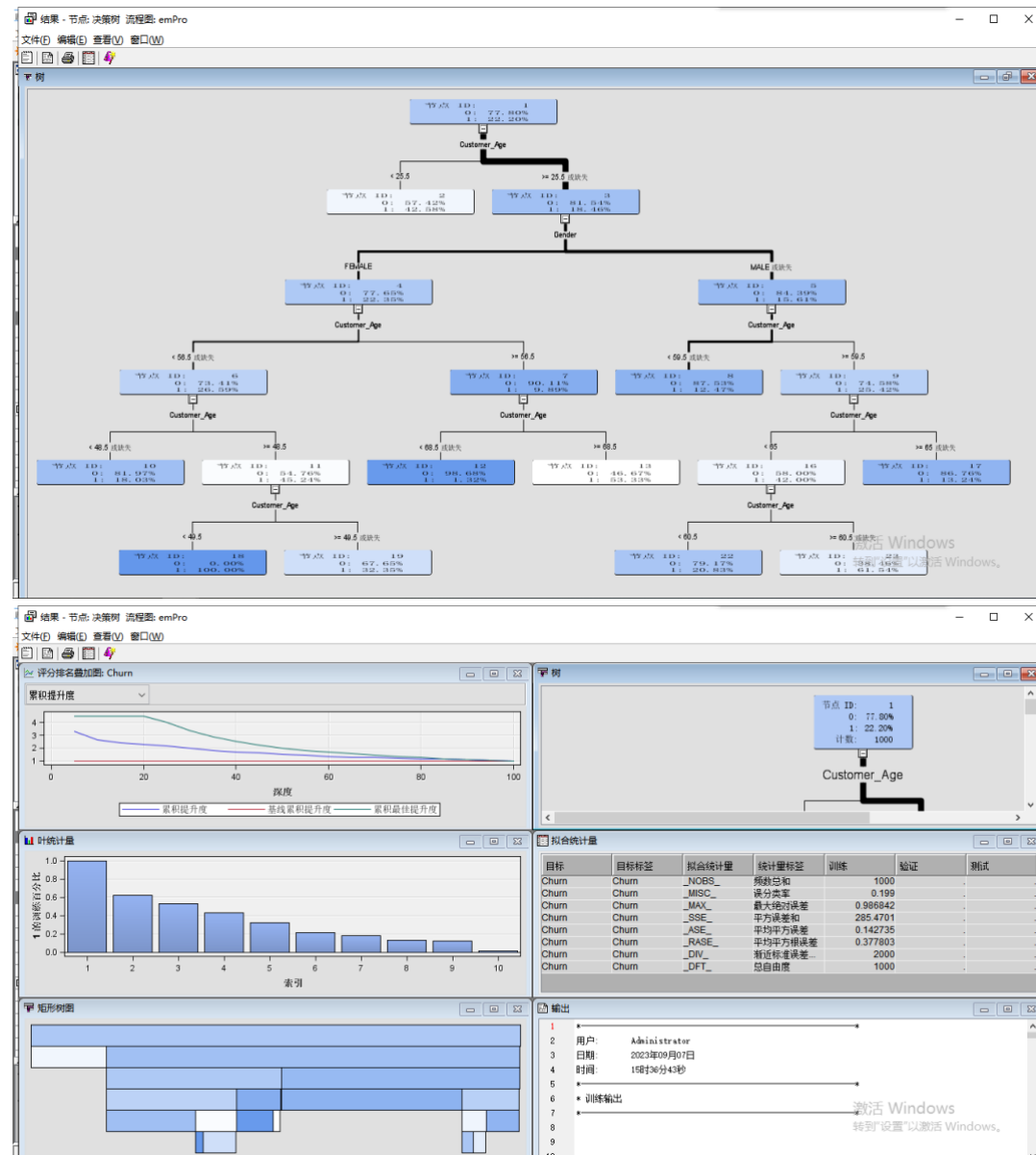
变量名称	补缺方法	补缺变量	补缺值	角色	测量水平	标签	TRAIN的缺失个数
Returns	DISTRIBUTION	IMP_Returns		INPUT	NOMINAL	Returns	179

Below the table, there's an '输出' (Output) pane showing the following information:

- 1 用户: Administrator
- 2 日期: 2023年09月07日
- 3 时间: 16:21:56
- 4
- 5
- 6 训练输出
- 7
- 8
- 9
- 10
- 11 变量汇总
- 12
- 13
- 14 角色 测量水平 频数计数
- 15
- 16

In the dataset, the dependent variable (target variable) should be "Churn" because this is the outcome I want to predict. All other variables can be used as independent variables (explanatory variables) to predict whether a customer will churn. Because there are two ages in the data. So I set up a rejection for one of them, because the name, ID, and date cannot be used for modeling, so I also set up a rejection.

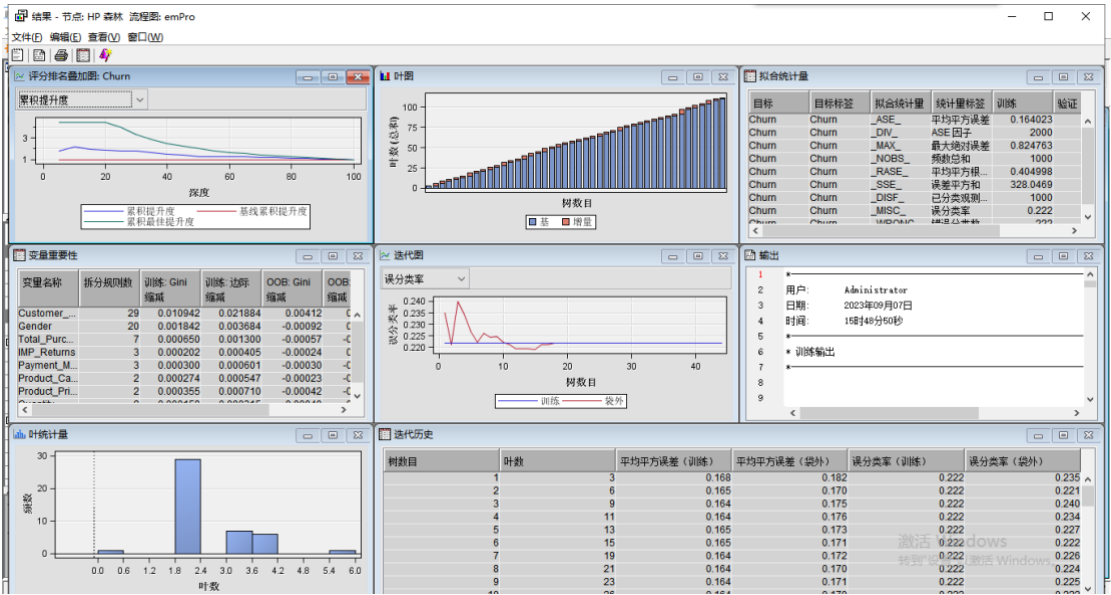
## Decision Tree:



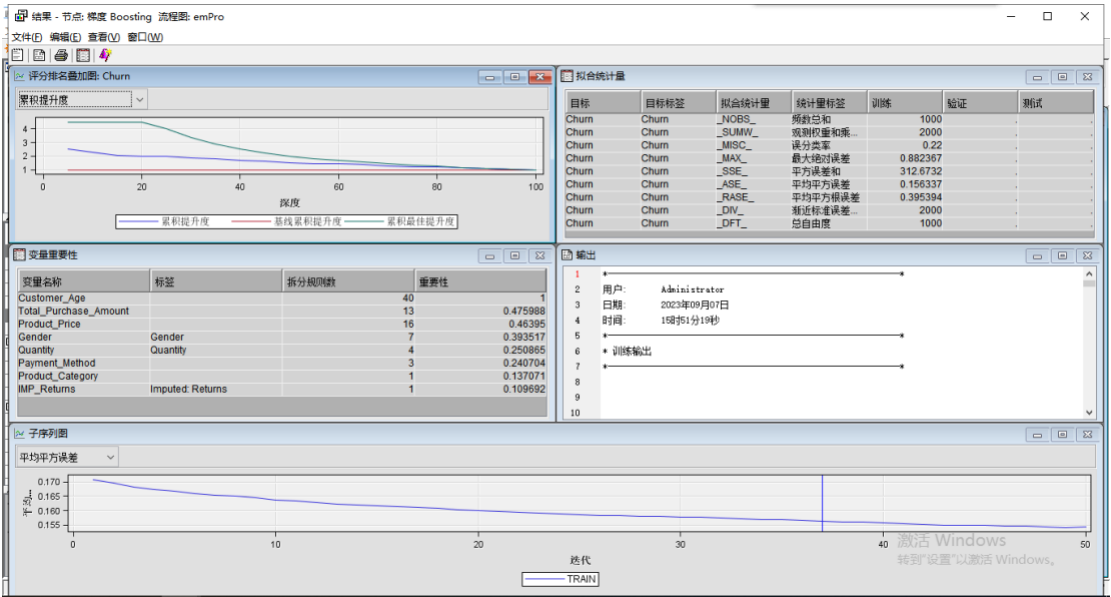
The problem is that an overfitted model relies too much on specific features in the training data and therefore may perform poorly on new, unknown data because those features may not apply to the new data. On the contrary, underfitting models perform poorly on both training and test data because they do not fully learn the characteristics of the training data and do not understand the data well enough. To solve these problems, we can use techniques such as limiting the depth of the decision tree or the minimum number of samples at a node to prevent overfitting, or consider using ensemble methods such as random forest or gradient boosting, which work by combining multiple models. Improve generalization

capabilities to achieve better performance on new data. These methods can effectively improve the performance of the model.

—Random Forest



—Boosting



These two models help items attempt to understand customer behavior. Identify which characteristics most affect customer churn, purchasing patterns, etc. At the same time, it can capture the complex nonlinear relationship between features and target variables. If customer behavior is influenced by the interactions between different variables, these methods can effectively model such effects. Given the previous analysis of decision trees and the construction of many unrelated trees and averaging their predictions, Random Forest is not easily overfitting. Boosting also has mechanisms to prevent overfitting, such as scaling down and random boosting, which involve training sub samples of the data.