

Business intelligence analytics experiment report

BUSINESS INTELLIGENCE ANALYTICS



**Subject:** Business Intelligence Analytics

**Professor:** Professor Lei Sun

Section: 2

**Members:**

Yu Jiang 22061300050

Longyue Yu 22061300015

Zehui Shi 22061300022

Shubo Lei 22061300099

Yiwen Liu 22061300016

**Date:** 12/31/2024

Contents

[1. Software: Python 1](#_Toc186577671)

[1.1. Software Introduction 1](#_Toc186577672)

[1.2. The main calling package 1](#_Toc186577673)

[2. Experiment 1: Prediction of telecom subscriber loss based on tree model 3](#_Toc186577674)

[2.1. Learning Objective 3](#_Toc186577675)

[2.1.1. Machine learning theory: Random forests and Adaboosting 3](#_Toc186577676)

[2.1.2. Experimental objective: Prediction of churn 3](#_Toc186577677)

[2.2. Data Resource 3](#_Toc186577678)

[2.2.1. Data Profile 3](#_Toc186577679)

[2.2.2. Data Dictionary 4](#_Toc186577680)

[2.3. Algorithm Introduction 5](#_Toc186577681)

[2.3.1. Random Forest 6](#_Toc186577682)

[2.3.2. Adaboosting 6](#_Toc186577683)

[2.4. Experimental process and Result Analysis 7](#_Toc186577684)

[2.4.1. Data Preprocessing 7](#_Toc186577685)

[2.4.2. Correlation Analysis 8](#_Toc186577686)

[2.4.3. Feature Screening 9](#_Toc186577687)

[2.4.4. Separate samples and oversample 11](#_Toc186577688)

[2.4.5. Test the model and adjust the parameters 11](#_Toc186577689)

[2.4.6. Build a model and analyze the results 13](#_Toc186577690)

[3. Experiment 2: Internet platform procurement intention forecast 15](#_Toc186577691)

[3.1. Learning Objective 15](#_Toc186577692)

[3.1.1. Machine learning theory: Artificial neural network 15](#_Toc186577693)

[3.1.2. Experimental objective: Prediction of Revenue 15](#_Toc186577694)

[3.2. Data Resource 15](#_Toc186577695)

[3.2.1. Data Profile 15](#_Toc186577696)

[3.2.2. Data Dictionary 15](#_Toc186577697)

[3.3. Algorithm Introduction 18](#_Toc186577698)

[3.3.1. Composition of ANN 18](#_Toc186577699)

[3.3.2. Backpropagation algorithm 18](#_Toc186577700)

[3.4. Experimental Process and Result Analysis 19](#_Toc186577701)

[3.4.1. Exploration of data sets 19](#_Toc186577702)

[3.4.2. Data Preprocessing 20](#_Toc186577703)

[3.5. Artificial Neural Network Modeling 23](#_Toc186577704)

[3.5.1. Preliminary Model 23](#_Toc186577705)

[3.5.2. Adjust Parameters: More Hidden Layers 24](#_Toc186577706)

[3.5.3. Adjust Parameters: Change Activation Function 25](#_Toc186577707)

[3.5.4. Train The Models and Compare Performance 25](#_Toc186577708)

[3.5.5. Prediction result analysis 26](#_Toc186577709)

[4. Experiment 3: Customer clustering analysis of a certain airline based on K-Means algorithm 27](#_Toc186577710)

[4.1. Learning Objective 27](#_Toc186577711)

[4.1.1. Machine learning theory: K-means 27](#_Toc186577712)

[4.1.2. Experimental objective: Airline Customer Value Analysis 27](#_Toc186577713)

[4.2. Data Resource 28](#_Toc186577714)

[4.2.1. Data Sources 28](#_Toc186577715)

[4.2.2. Data Dictionary 28](#_Toc186577716)

[4.3. Algorithm Introduction 30](#_Toc186577717)

[4.3.1. K – means Algorithm 30](#_Toc186577718)

[4.3.2. RFM Model 30](#_Toc186577719)

[4.3.3. LRMFC Model 31](#_Toc186577720)

[4.4. Experimental process and Result Analysis 32](#_Toc186577721)

[4.4.1. Data Standardization 32](#_Toc186577722)

[4.4.2. Establish K-Means clustering model 32](#_Toc186577723)

[4.4.3. Optimal k-value selection and visualization 33](#_Toc186577724)

[4.4.4. Customer Value analysis 33](#_Toc186577725)

[5. Experiment 4: Product Recommendation Based on Association Rules 35](#_Toc186577726)

[5.1. Learning Objective 35](#_Toc186577727)

[5.1.1. Data mining: slicing, dicing, and visualization 35](#_Toc186577728)

[5.1.2. Machine Learning Theory: Association Rules 35](#_Toc186577729)

[5.2. Data Source 35](#_Toc186577730)

[5.3. Introduction of Apriori Algorithm 36](#_Toc186577731)

[5.3.1. Three Important Indicators of Association Rules 36](#_Toc186577732)

[5.3.2. Apriori Principle (Antimonotonicity) 36](#_Toc186577733)

[5.3.3. Apriori Procedure 36](#_Toc186577734)

[5.3.4. Generate all Association Rules 37](#_Toc186577735)

[5.4. Experimental Process and Result Analysis 37](#_Toc186577736)

[5.4.1. Data Preprocessing: Data Transformation of Time Type 37](#_Toc186577737)

[5.4.2. Data exploration: Descriptive statistics and visualization 38](#_Toc186577738)

[5.4.3. Dice: Time and Item 41](#_Toc186577739)

[5.4.4. Generation of Association Rule 42](#_Toc186577740)

[Appendix 47](#_Toc186577741)

[Raw data extract 47](#_Toc186577742)

[Experiment 1 47](#_Toc186577743)

[Experiment 2 47](#_Toc186577744)

[Experiment 3 47](#_Toc186577745)

[Experiment 4 47](#_Toc186577746)

Contents of Figures

[Figure 1 Correlation Matix 9](#_Toc186577747)

[Figure 2 Feature Importances Bar Chart 10](#_Toc186577748)

[Figure 3 Learning Curve for n\_estimators of Ada 13](#_Toc186577749)

[Figure 4 CM of RF and Ada 13](#_Toc186577750)

[Figure 5 Roc Curve of RF and Ada 14](#_Toc186577751)

[Figure 6 Artificial neural network schematic 18](#_Toc186577752)

[Figure 7 Revenue Bar Chart 20](#_Toc186577753)

[Figure 8 classifier schematic diagram 24](#_Toc186577754)

[Figure 9 classifier1 schematic diagram 25](#_Toc186577755)

[Figure 10 Relationship diagram between contour coefficient and k value 32](#_Toc186577756)

[Figure 11 Customer Value Clustering Line Chart 33](#_Toc186577757)

[Figure 12 Top 20 ltems purchased 38](#_Toc186577758)

[Figure 13 Number of orders received each month 38](#_Toc186577759)

[Figure 14 Number of orders received each day. 39](#_Toc186577760)

[Figure 15 Count of orders received each hour 41](#_Toc186577761)

[Figure 16 Count of orders received each periodof a day 41](#_Toc186577762)

[Figure 17 Top 10 most sold products in each period 42](#_Toc186577763)

[Figure 18 Top 10 most sold products in each season 42](#_Toc186577764)

Contents of Tables

[Table 1 Experiment 1: Data Dictionary 4](#_Toc186577765)

[Table 2 Data head after One-Hot 8](#_Toc186577766)

[Table 3 Data Head after Standardization 8](#_Toc186577767)

[Table 4 Features used for RF and Ada 11](#_Toc186577768)

[Table 5 Parameter Rangs for GridSearch 12](#_Toc186577769)

[Table 6 Performace of RF and Ada 14](#_Toc186577770)

[Table 7 Experiment 2: Data Dictionary 15](#_Toc186577771)

[Table 8 Data Description of Experiment 2 19](#_Toc186577772)

[Table 9 Correlation Matrix of Features 20](#_Toc186577773)

[Table 10 Data Information after One-Hot 22](#_Toc186577774)

[Table 11 Data Description after Standardization 23](#_Toc186577775)

[Table 13 Comparison of 3 ANN Models 26](#_Toc186577776)

[Table 14 Experiment 3: Data Dictionary 28](#_Toc186577777)

[Table 15 Standardized Data 32](#_Toc186577778)

[Table 16 Customer Value Analysis 34](#_Toc186577779)

[Table 17 Experiment 4:Data Dictionary 35](#_Toc186577780)

[Table 18 Raw Data Head 37](#_Toc186577781)

[Table 19 Data Head after Coding 37](#_Toc186577782)

[Table 20 Daily Transaction over 200 40](#_Toc186577783)

[Table 21 Transaction Table 43](#_Toc186577784)

[Table 22 Frequent Items 43](#_Toc186577785)

[Table 23 Association Rules 45](#_Toc186577786)

# Software: Python

## Software Introduction

In this experiment, we chose to use python software for analysis and processing. ‌Python is an interpreted, object-oriented programming language known for its simplicity, ease of reading, and ease of maintenance. The syntax is designed to emphasize code readability and allow complex logic to be expressed in less code.

python has a wide range of libraries and frameworks, with rich data analysis and machine learning libraries, providing powerful data processing capabilities and model construction tools.

At the same time, python's flexibility and scalability make it possible to handle problems of all sizes, and it can easily integrate with other systems and languages to adapt to changing data analysis needs, whether it is the growth of data size or the introduction of new algorithms.

Python's data visualization capabilities provide data visualization libraries such as Matplotlib and Seaborn, which help users create intuitive charts and graphs to better understand data.

Object-oriented learning in Python can improve code organization and modularity, make management easier, enhance the encapsulation of data and methods, protect data and methods from incorrect use or modification by outsiders, implement polymorphism, simplify interface processing, and allow a unified interface to handle different types of objects.

## The main calling package

Data analysis and processing

* pandas as pd
* numpy as np

Data visualization

* matplotlib.pyplot as plt
* seaborn as sns

Data preprocessing

* SMOTE
* datetime
* time

Machine Learning sklearn - Model Selection and Evaluation:

* model\_selection
* preprocessing
* LabelEncoder
* OneHotEncoder
* StandardScaler
* metrics
* confusion\_matrix
* accuracy\_score

Machine Learning sklearn - Integrated Learning Approach:

* RandomForestClassifier
* AdaBoostClassifier
* ExtraTreesClassifier
* GradientBoostingClassifier

Deep learning Framework:

* tensorflow.keras as keras
* Sequential
* Dense
* Dropou

Clustering algorithm

* KMeans

Association rule mining:

* association\_rules
* apriori

# Experiment 1: Prediction of telecom subscriber loss based on tree model

## Learning Objective

### Machine learning theory: Random forests and Adaboosting

Through two integrated algorithms based on tree models, we expect to grasp the general flow of machine learning tasks, including data preprocessing, feature screening, parameter optimization, and model evaluation.

Through the process of parameter optimization and adjustment, we should understand the basic principle and framework of the integrated algorithm, as well as the influence of important parameters of the tree model on the model fitting.

At the same time, we expect to learn how to use the python sklearn library through practical operations, including class calls and instantiations, grid search, important interface calls, and model evaluation methods.

### Experimental objective: Prediction of churn

We want to use the data set of telecom user loss to establish a classification model of user loss, get the key characteristics that affect user loss, predict the future user loss situation, and formulate early warning and recall strategies for potential user loss.

## Data Resource

### Data Profile

The data file originates from WA\_Fn-UseC\_-Telco-Customer-Churn.csv. This dataset records various information about customers from a telecommunications company in the United States, including personal information, service usage, billing information, and whether they will churn (Churn). The dataset contains 7043 records and 21 fields.

### Data Dictionary

Table 1 Experiment 1: Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Description | Data Type | Possible Values |
| customerID | Unique identifier for the customer | String | Unique ID for each customer |
| Churn | Whether the customer has churned, the target variable | Binary classification | 0: Not Churned, 1: Churned |
| gender | Customer's gender | Categorical variable | 0: Female, 1: Male |
| SeniorCitizen | Whether the customer is a senior citizen | Binary classification | 0: No, 1: Yes |
| Partner | Whether the customer has a partner | Binary classification | 0: No, 1: Yes |
| Dependents | Whether the customer has dependents | Binary classification | 0: No, 1: Yes |
| tenure | The duration of the customer's service with the company (in months) | Continuous variable | Numerical values indicating the length of service with the company |
| PhoneService | Whether the customer has subscribed to telephone services | Binary classification | 0: No, 1: Yes |
| MultipleLines | Whether the customer has subscribed to multiple lines | Binary classification | 0: No, 1: Yes, -1: No phone service |
| InternetService | Whether the customer has subscribed to internet services | Categorical variable | 0: DSL,  1: Fiber optic,  2: No internet service |
| OnlineSecurity | Whether the customer has subscribed to online security services | Binary classification | 0: No, 1: Yes,  NaN: Missing |
| OnlineBackup | Whether the customer has subscribed to online backup services | Binary classification | 0: No, 1: Yes |
| DeviceProtection | Whether the customer has subscribed to device protection services | Binary classification | 0: No, 1: Yes |
| TechSupport | Whether the customer has subscribed to technical support services | Binary classification | 0: No, 1: Yes |
| StreamingTV | Whether the customer has subscribed to streaming TV services | Binary classification | 0: No, 1: Yes |
| StreamingMovies | Whether the customer has subscribed to streaming movie services | Binary classification | 0: No, 1: Yes |
| Contract | The type of contract the customer has | Categorical variable | 0: Month-to-month, 1: One year, 2: Two year |
| PaperlessBilling | Whether the customer has opted for paperless billing | Binary classification | 0: No, 1: Yes |
| PaymentMethod | The customer's payment method | Categorical variable | 0: Electronic check, 1: Mailed check,  2: Bank transfer (automatic),  3: Credit card (automatic) |
| MonthlyCharges | The customer's monthly charges | Continuous variable | Numerical values indicating the monthly cost |
| TotalCharges | The customer's total charges | Continuous variable | Numerical values indicating the total cost |

## Algorithm Introduction

Ensemble algorithm refers to a machine learning method which builds multiple base learners and combines the results of base learners through certain strategies to get the final result. According to different combination strategies, it can be divided into bagging, boosting and stacking.

In order to ensure that the integrated algorithm can get better results than the base learner, the requirements of the integrated algorithm for the base learner are:

1. Can't be too bad: The base model is correct at least better than a random guess (50%)

2. Be diverse: Not too similar

Random Forest (RF) and Adaboosting are both integrated algorithms based on tree models, and their base learners are decision trees.

### Random Forest

The combination strategy of Bagging is voting or averaging. Random forest is the best known bagging algorithm, which creates diversity of base learners through random attributes and bootstrap sampling (random sampling with put back).

For a random forest classifier with n trees (n\_estimators = n), a sample size of z, and a maximum of m features per tree(max\_features = m), the algorithm is as follows:

for i in range(1: n):

Bootstrap z times from the training set of the random forest to form the training set of the specific tree.

Construct decision tree DTi and get the result of the classification task

Vote based on the results of n trees to get the final result of the classification task

Among them, the algorithm for building each decision tree is as follows:

Randomly select m of the features to get the characteristic variables of this tree

while(stopping criteria are not met):

Calculate and sort the information gain(or other criteria) for each of the rest feature variable

Split the feature with the greatest information gain

### Adaboosting

The core of Boosting algorithm is to increase the weight of predicted failed samples with heavier learning tasks last time, so as to continuously improve the prediction ability of learners and transform weak learners into strong learners.

Adaboosting is the most basic boosting algorithm. It creates the independence of the base learner by Re-weighting and Re-sampling. The algorithm is as follows:

#：training set with N data，in which features variable is  and target variable is

#：base learner

#：a super parameter that denotes the number of base learners

#：The weight of each sample in the training set

# The initial weights for each sample are equal

for t in range(1: ):

if : choose next classifier

else:

for n in range(1:N):

if :

if :

# Renormalizing weights

## Experimental process and Result Analysis

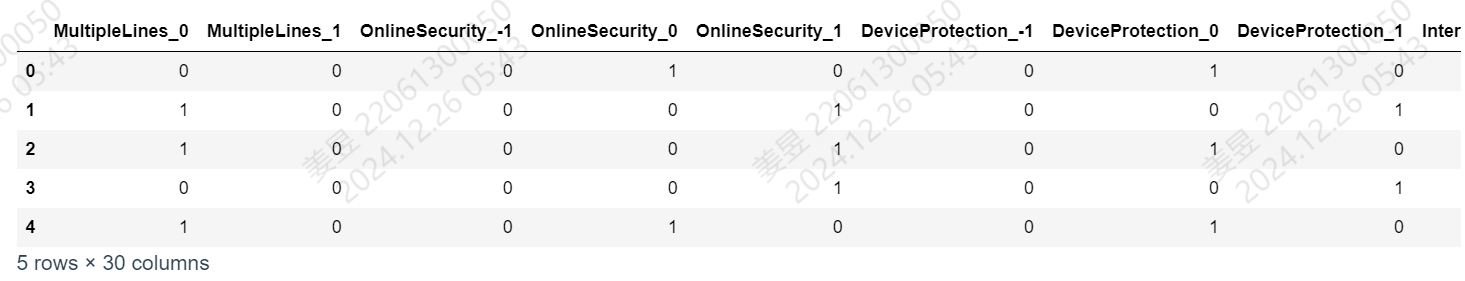
### Data Preprocessing

For the character variable in the sample, we convert it to a numeric variable. Consider 'MultipleLines', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies' is actually a detailed case of 'PhoneService' and 'InternetService', and we encode both No phoneservice and No internetservice as 0.

At the same time, we noticed that there were 11 user data missing from the 'TotalCharges'. By observation, these 11 people have tenure=0. This indicates that these 11 people have just applied for the package and have used it for less than one month, so we use the value of "MonthlyCharges" to fill in.

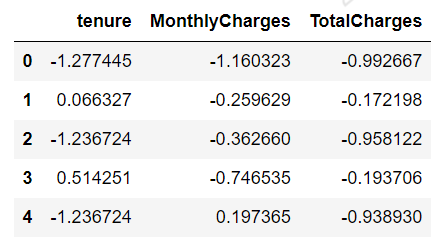
For the categorical variables in the sample, we adopt unique heat coding. Using the get\_dummies method in the pandas library, convert each possible value of a categorical variable into a binary vector. Unique heat coding makes it easier for models to learn and interpret features, and ensures that each category has the same weight, without numerical differences in the original data affecting the model's judgment of the importance of features. The data head after One-Hot coding is show as Table 2.

Table 2 Data head after One-Hot



For the continuous variables' tenure, 'MonthlyCharges',' TotalCharges', we use preprocessing. StandardScaler standardized () class in the sklearn library. This method uses the mean and standard deviation of continuous variables for normalization. Standardization can unify dimensions and ensure the same feature weights, while helping to speed up convergence and improve model accuracy, while numerical instability creates the risk of gradient explosion. The result after conversion is shown in Table 3.

Table 3 Data Head after Standardization



### Correlation Analysis

We plotted the correlation heat maps of features and targets, as shown in Figure 1.

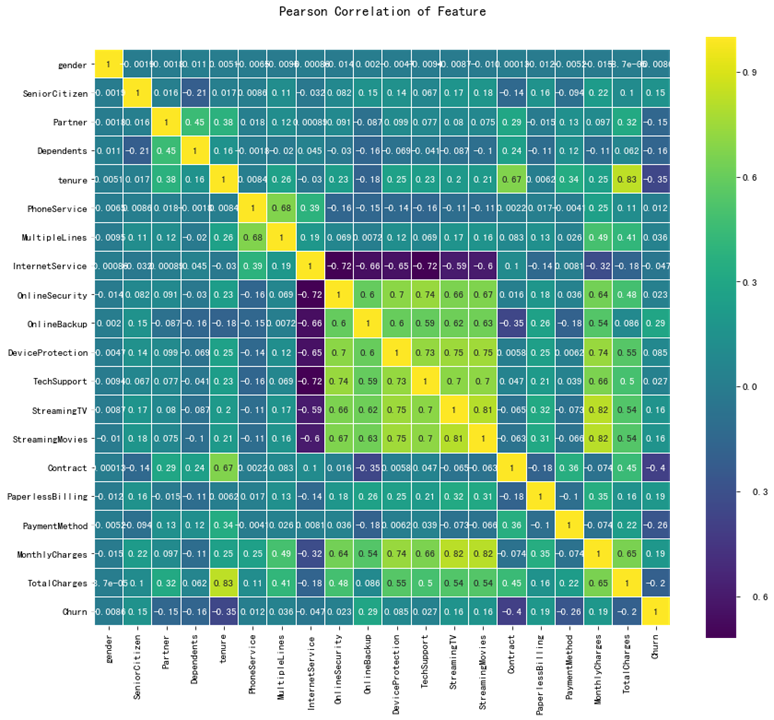


Figure 1 Correlation Matix

Feature correlation analysis can help us understand how much each feature influences model predictions, which is important for explaining model decision processes and trusting model outputs.

### Feature Screening

Considering the large number of features in the sample, it is easy to cause overfitting, so we first screened the features.

First, we calculate the top 20 features and corresponding importances of features\_importance\_ in the grid search results of the two models. It is stored in feature\_imp\_sorted\_rf and feature\_imp\_sorted\_ada. These 40 features are then de-weighted and the average value of the importance of repeated features is used as the overall average value and stored in imp\_all. We have drawn the following bar graph as Figure 2.

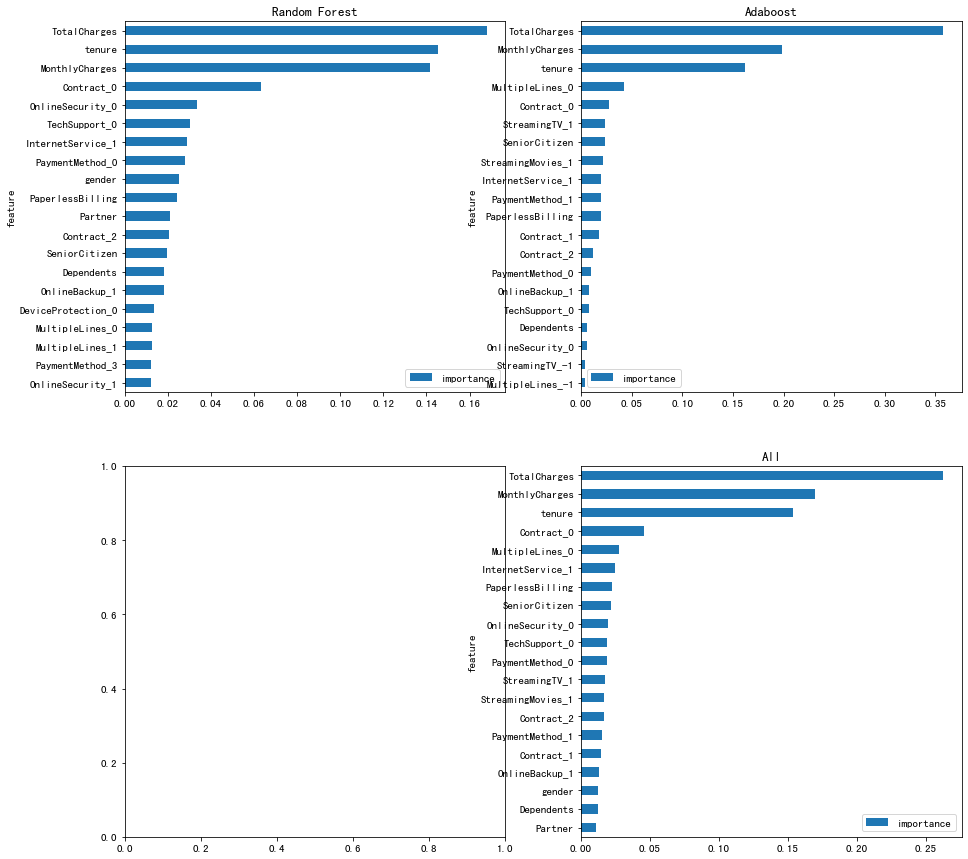
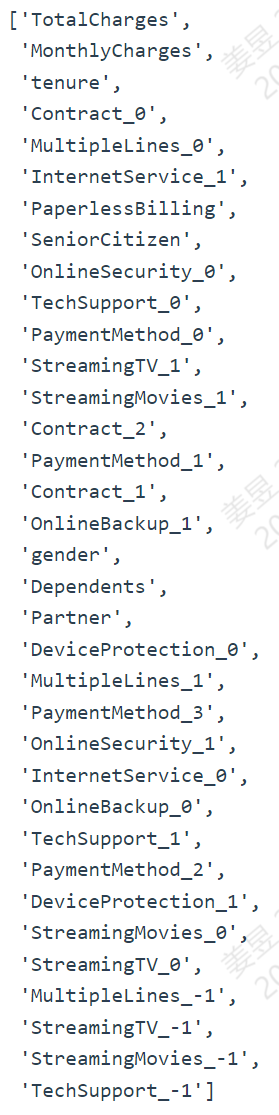


Figure 2 Feature Importances Bar Chart

As can be seen from the figure, the user's total payment, monthly payment, time already in use, contract payment method, and which type of network service features are more important.

Finally, we selected the first 35 features in imp\_all as input to the model. These features are shown in Table 4.

Table 4 Features used for RF and Ada



### Separate samples and oversample

We used the sklearn library's model\_selection.train\_test\_split to separate the training set from the test set, using a test ratio of 0.2.

Given that the distribution of Churn for the target variable is uneven, the model has a preference for negative samples when making direct predictions, which results in a low recall rate. Therefore, SMOTE oversampling algorithm was used to balance the data set when generating the training set and test set.

### Test the model and adjust the parameters

We built a random forest model using sklearn RandomForestClassifier, and used model\_selection.GridSearchCV for parameter optimization. The adjustment range of the parameters we set is shown in Table 5.

Table 5 Parameter Rangs for GridSearch

|  |  |
| --- | --- |
| Parameter | Range |
| n\_estimators | [500, 600, 700, 800, 900, 1000] |
| max\_depth | [6, 8, 10, 12, 15, 20] |
| min\_samples\_split | range(10, 90, 20) |
| min\_samples\_leaf | range(5, 65, 10) |
| max\_features | 'sqrt','auto','log2' |
| warm\_start | False, True |

The best parameter combination after grid search is:

'max\_depth': 15,

'max\_features': 'sqrt',

'min\_samples\_leaf': 5,

'min\_samples\_split': 10,

'n\_estimators': 700,

'warm\_start': True.

The corresponding best\_score of random forest is: 0.8170530638309195

We built the AdaBoosting model using sklearn's AdaBoostClassifier. Considering that the AdaBoostClassifier in sklearn library has few important parameters, we use the learning curve to optimize the parameters of n\_estimators. We have made the learning curve of n\_estimators=range(100, 1500,100) for 15 cases, as shown in the figure below:

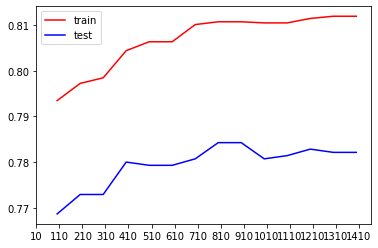


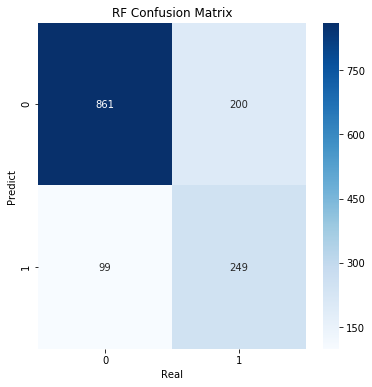
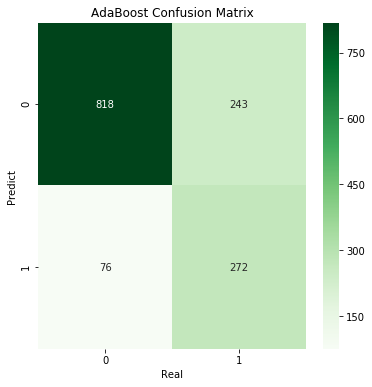
Figure 3 Learning Curve for n\_estimators of Ada

As can be seen from the figure, the optimal n\_estimators is: 800

At this point, the score of the AdaBoostClassifier on the test set is 0.7842441447835344

### Build a model and analyze the results

Figure 4 CM of RF and Ada



Comparing the two confusion matrices shown in Figure 4, it can be seen that the prediction ability of random forest is better than that of Adaboosting when it is actually negative, while the prediction ability of random forest is worse than that of Adaboosting when it is actually positive.

We established the training model of the two models under the optimal parameters, and calculated the Accuracy and Recall rate on the test set, as shown in Table 6.

Table 6 Performace of RF and Ada

|  |  |  |
| --- | --- | --- |
| Model | RF | Adaboosting |
| Accuracy | 0.78637 | 0.7736 |
| Recall | 0.71264 | 0.78161 |

According to Table 5, although random forest has a higher accuracy, its recall is very low. Considering the actual background and data characteristics of the experiment, the Adaboosting model with higher recall and high accuracy is more suitable for this prediction task.

We also plotted ROC curves and confusion matrices for the two models, as shown in Figure 5.

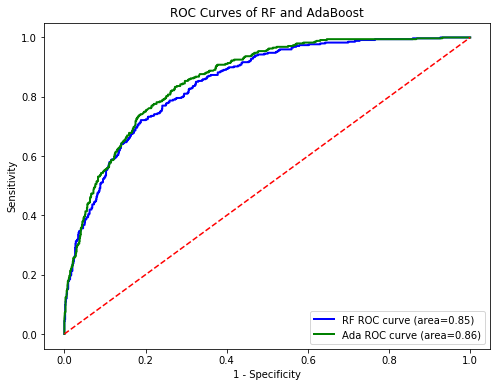


Figure 5 Roc Curve of RF and Ada

As can be seen from Figure 5, although ROC curves of the two models were very close, Adaboosting had better predictive ability than RandomForest when Specificity was high.

In summary, we believe that Adaboosting performs better in this prediction task.

# Experiment 2: Internet platform procurement intention forecast

## Learning Objective

### Machine learning theory: Artificial neural network

By exploring and modeling a large dataset, we hope to understand the general flow of artificial neural networks implemented in python, as well as the main parameters of artificial neural networks (number of hidden layers and activation function).

### Experimental objective: Prediction of Revenue

Using the data set of user access records of an Internet platform, we hope to achieve the prediction of user purchase behavior, and use the prediction results to adopt different marketing methods for different customer groups.

Specifically, based on the prediction with high accuracy, once we are able to determine that someone will generate revenue, we do not need to offer any coupons and can offer visitors special credits that they can use the next time they visit, encouraging multiple repeat purchases; And for those visitors who are less likely to buy, we will get discount coupons so that they are more likely to buy.

## Data Resource

### Data Profile

The data used in this experiment comes from the user access record dataset of a certain platform, which contains 12,330 pieces of data, each piece of data represents a user's visit, and each access data contains 17 characteristic values, including 10 continuous variables and 7 discrete variables.

Our goal is to predict **Revenue**, which is a Boolean variable, meaning whether users place orders, that is, whether browsing generates revenue.

### Data Dictionary

Table 7 Experiment 2: Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Description | Data Type | Unit or Possible Values |
| Administrative | Number of account management-related pages viewed by the user | int64 | pages |
| Administrative\_Duration | Time spent on account management pages | float64 | secondes |
| Informational | Number of informational pages viewed by the user (e.g., contact information) | int64 | pages |
| Informational\_Duration | Time spent on informational pages | float64 | secondes |
| ProductRelated | Number of product-related pages viewed by the user | int64 | pages |
| ProductRelated\_Duration | Time spent on product-related pages | float64 | secondes |
| BounceRates | Bounce rate | float64 | 0-100 |
| ExitRates | Exit rate | float64 | 0-100 |
| PageValues | The value of the goods on the page | float64 | 0-361.763742 |
| SpecialDay | Whether the visit date is close to a special day such as a holiday | float64 | 0-1 |
| Month | Visit month | object | Feb', 'Mar', 'May', 'Oct', 'June', 'Jul', 'Aug', 'Nov', 'Sep', 'Dec' |
| OperatingSystems | Operating system | int64 | 1,2,3,4,5,6,7,8 |
| Browser | Browser | int64 | 1,2,3,4,5,6,7,8,9,10,11,12,13 |
| Region | Region | int64 | 1,2,3,4,5,6,7,8,9 |
| TrafficType | Source of traffic | int64 | int from 1 to 20 |
| VisitorType | Visitor type | object | Returning\_Visitor', 'New\_Visitor', 'Other' |
| Weekend | Whether the visit time was during the weekend | bool | 0 = No, 1= Yes |
| Revenue | Whether an order was placed | bool | 0 = No, 1= Yes |

As can be seen from the Table 7, some categorical variables are labeled as numbers, and the meanings of different labels are missing. In this case, the explanatory power of the model may not matter much, because even a highly explanatory model (such as a decision tree) cannot explain the results.

## Algorithm Introduction

### Composition of ANN

The design of artificial neural networks is inspired by biological neural networks, specifically the way the human brain works. It attempts to simulate the interactions and information processing between neurons in the brain.

The basic elements of an artificial neural network include nodes and connections, each of which includes weights and bias, and nodes use activation functions. A neural network consists of multiple layers of neurons, including an input layer, one or more hidden layers, and an output layer. Figure 6 is a schematic of an ANN model.

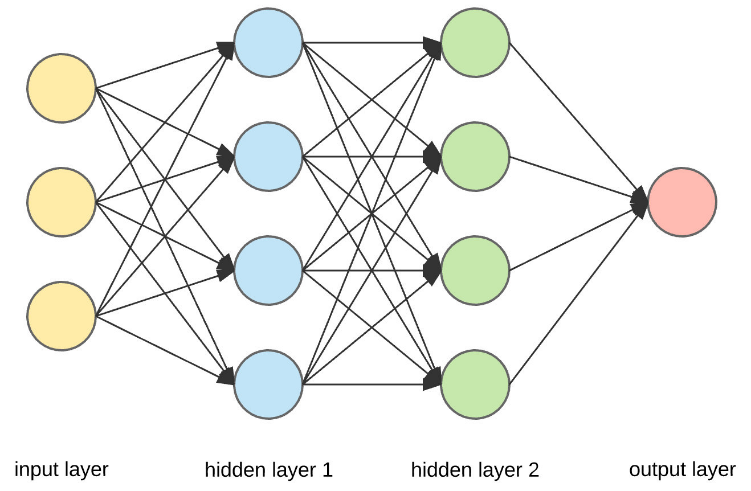


Figure 6 Artificial neural network schematic

Its working principle includes forward propagation and back propagation. In forward propagation, information flows from the input layer to the output layer to output the prediction result. Backpropagation: Reduce prediction errors by calculating errors in the output layer and assigning these errors to the neurons in each layer in proportion to the weight to adjust the weight and bias.

### Backpropagation algorithm

Next I will show how the backpropagation algorithm optimizes the parameters.

For training set

Determine the number of hidden layers of the model and the number of nodes per layer.

Determine the learning rate .

Randomly initializes all connection weights and bias in the network.

repeat:

for i = L, L-1, L-2, …, 1

calculate gradient of the loss function to the parameters of the current layer , denoted

Update parameter  according to

Update parameter in a similar way

if the stop criteria is not met, break

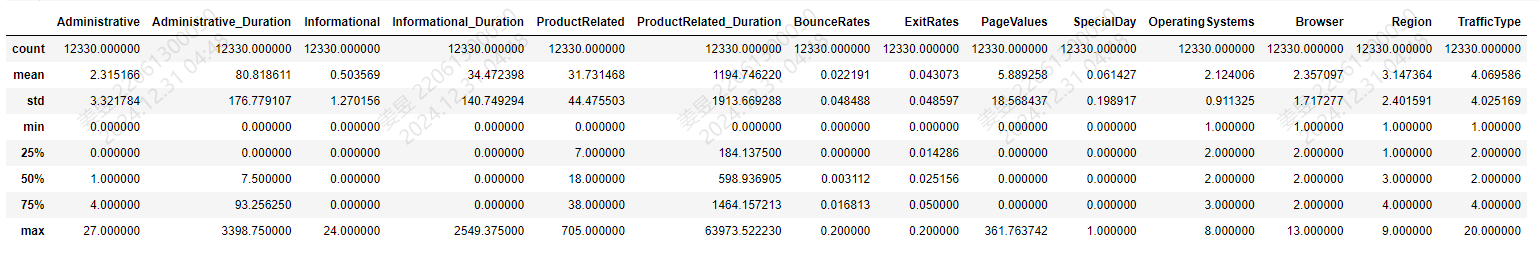
## Experimental Process and Result Analysis

### Exploration of data sets

First, we use the describe() function to view descriptive statistics for each feature and tag, shown in Table 8.

As can be seen from the results, the mean values of each column differ greatly from each other, so the range of features should be normalized.

Table 8 Data Description of Experiment 2



After dividing the data into features and labels, we draw a bar chart for label Revenue, as shown in Figure 7.

As can be seen from the figure, the prediction label is highly unbalanced, with negative far exceeding positive.

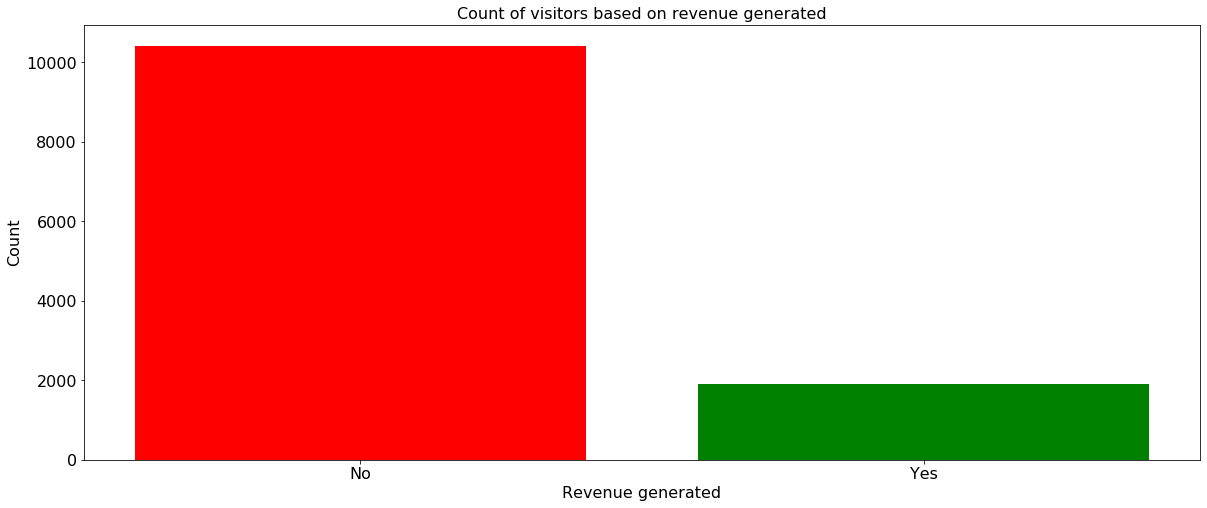


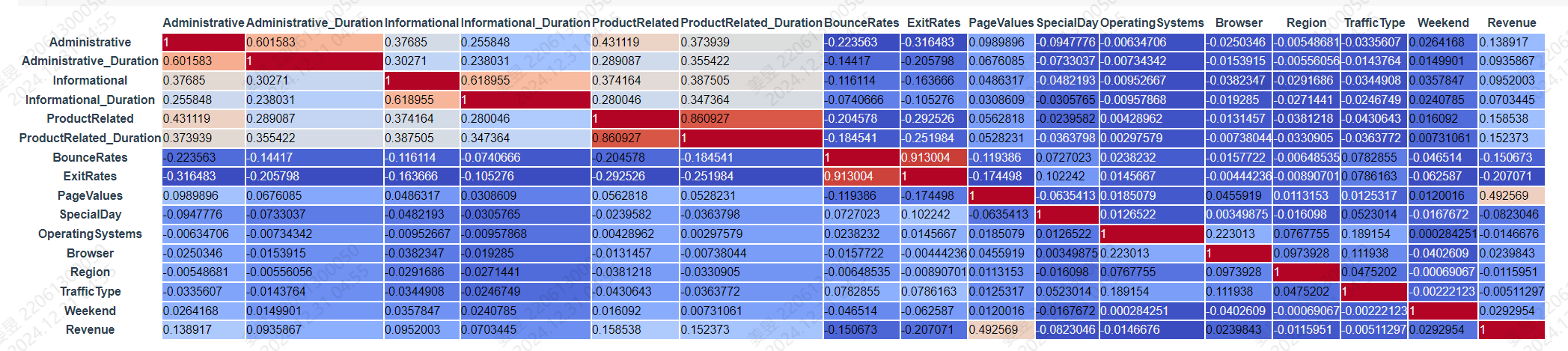
Figure 7 Revenue Bar Chart

### Data Preprocessing

#### Data Cleaning

We drew the correlation matrix of each feature and label, and set the shading color with coolwarm cmap, as shown in Table 9.

Table 9 Correlation Matrix of Features



As can be seen from Table 9, PageValues has the largest linear correlation with our target value, with a linear correlation coefficient of 0.492569.

Considering that there are many characteristic variables, we consider deleting some variables with little correlation.

The correlation between the feature variables OperatingSystems, Region, and TrafficType and the tag Revenue is less than 0.02, and they have little impact on the Revenue, so we delete these attributes directly.

In addition, Administrative and administrative \_duration, The relationships between Informational and Informational\_Duration and ProductRelated and ProductRelated\_Duration seem to have very high correlations, with very high collinearity. This data-illustrated phenomenon is consistent with immediacy and common sense, since the duration of time spent on a certain type of page is certainly affected by the number of page types visited. Therefore, we delete the number of pages visited for each type and use Duration to represent the visit time and the number of pages visited.

#### Spilt Training vs Test

We used the sklearn library's model\_selection.train\_test\_split to separate the training set from the test set, using a test ratio of 0.2.

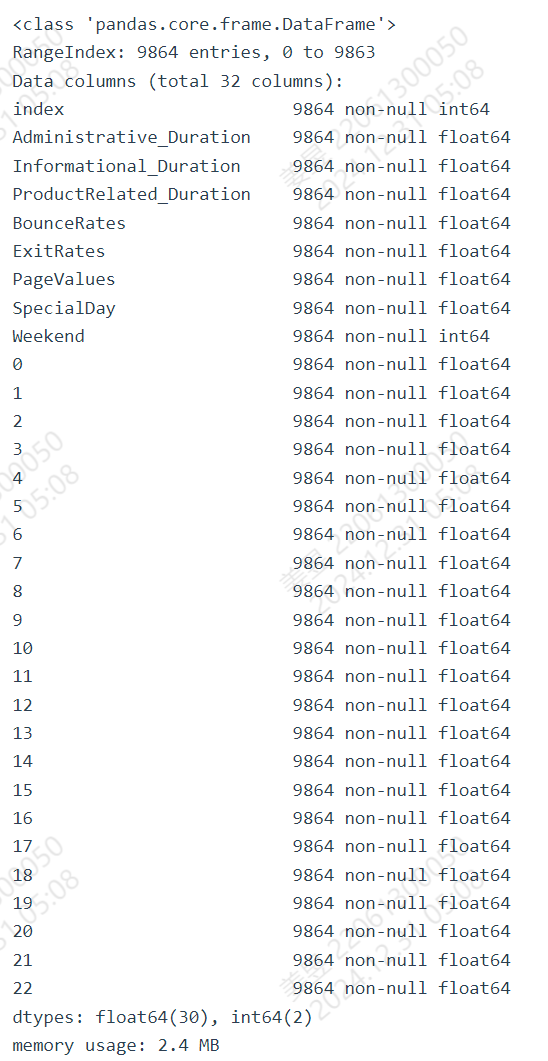
#### One-Hot Coding

In artificial neural networks, using the unique thermal coding can avoid the bias introduced by the size of the feature value, and use the sparse vector generated by the unique thermal coding directly as the input, so as to improve the prediction ability of the model. So we use LabelEncoder to convert the labels of categorical variables to numeric variables, and OneHotEncoder to create separate columns for each class in the column.

We removed the original eigenvector.

Given that ANN is a black box algorithm with poor interpretation, naming features generated by unique thermal coding is of little significance, so we retain the feature names generated by OneHotEncoder (natural numbers 0-22). The encoded training set information is shown in Table 10.

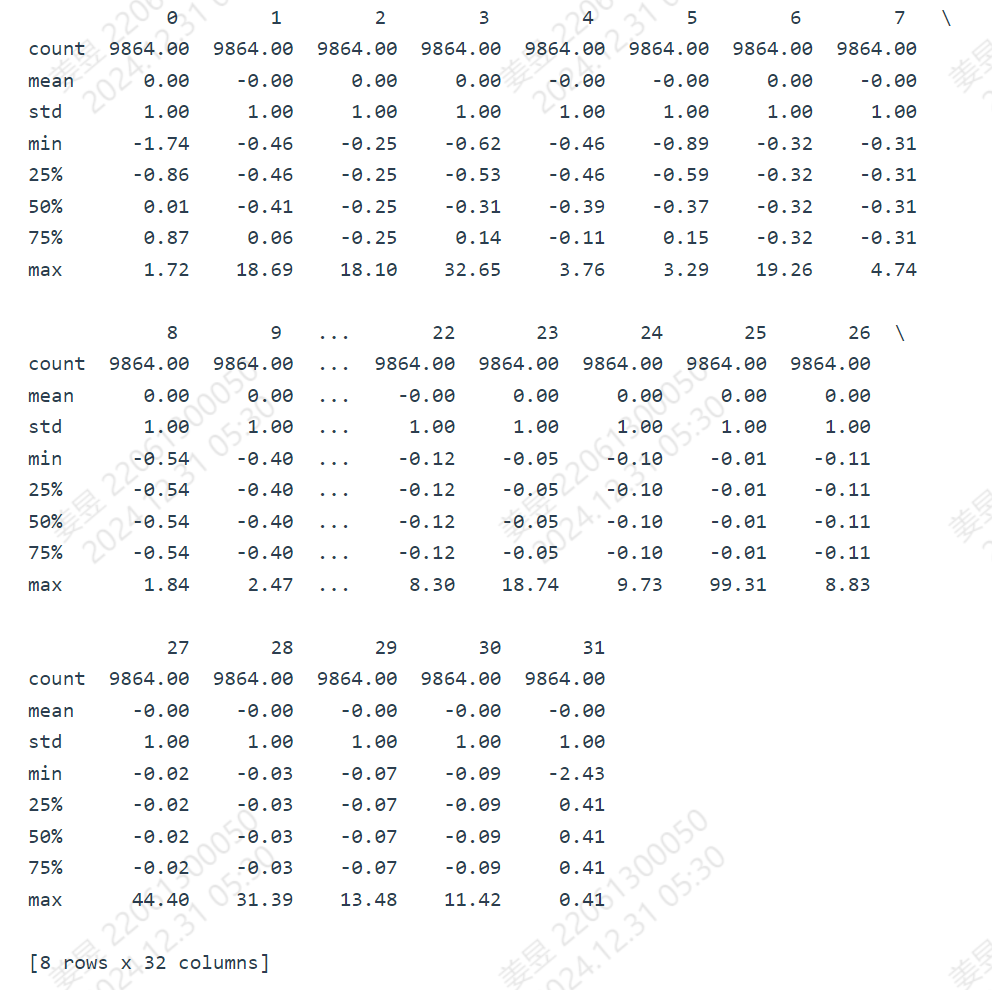
Table 10 Data Information after One-Hot



#### Scaling

We used StandardScaler() to standardize the eigenmatrix of the training set and the test set to make the data conform to a normal distribution, and the descriptive statistics of the processed data were shown in Table 11 (Using lambda functions to limit all numbers to 2 decimal places).

Table 11 Data Description after Standardization



## Artificial Neural Network Modeling

### Preliminary Model

We constructed a preliminary deep neural network model classifier using the Keras library.

The model starts with an input layer of 128 neurons and contains 3 hidden layers. Each hidden layer is a fully connected layer Dense and one by one Dropout layer to reduce overfitting. The ‘ReLU’ activation function is used in the hidden layer.

The output layer of the model contains a neuron that uses the sigmoid’ activation function to output a probability value between 0 and 1, representing the probability that the sample belongs to a positive class.

The model was trained using the ‘adam’ optimizer, the loss function is ‘binary\_crossentropy’, and the evaluation index is ‘accuracy’.

Figure 8 is a schematic diagram of classifier

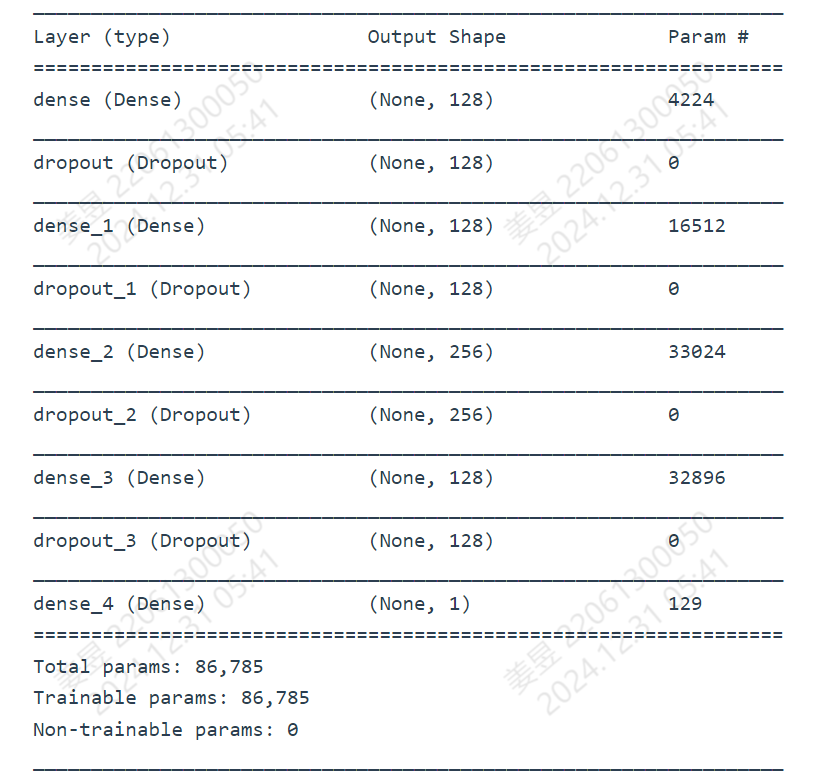


Figure 8 classifier schematic diagram

### Adjust Parameters: More Hidden Layers

Keeping other parameters unchanged, we increased the number of hidden layers to 5 layers, and established the model classifier 1. The diagram is shown in Figure 9.

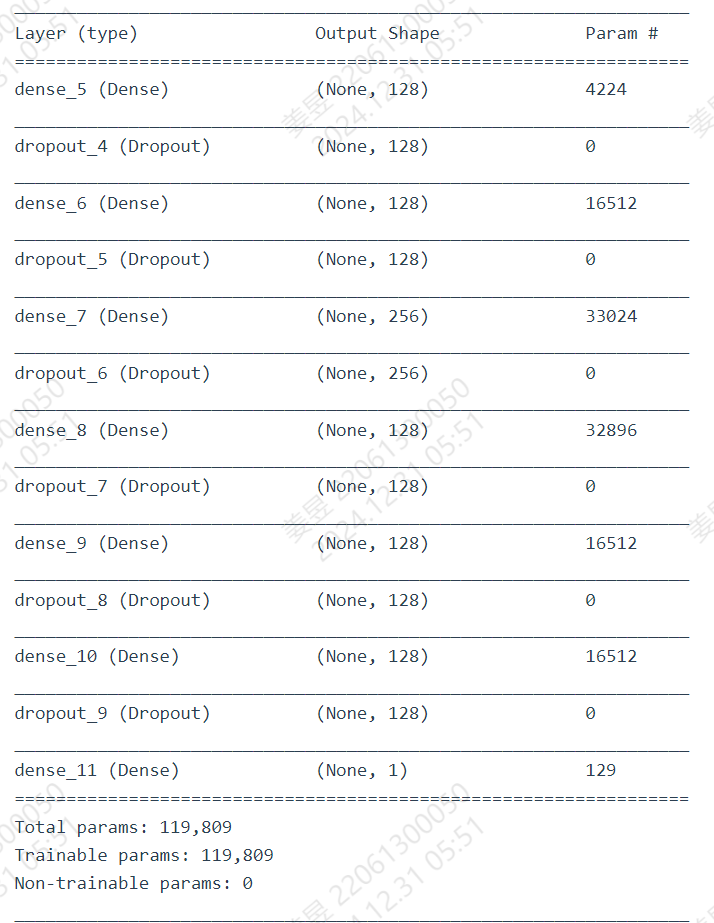


Figure 9 classifier1 schematic diagram

### Adjust Parameters: Change Activation Function

Based on classifier, we adjusted the activation function used by the input layer and the hidden layer to sigmoid, and established the model classifier 2.

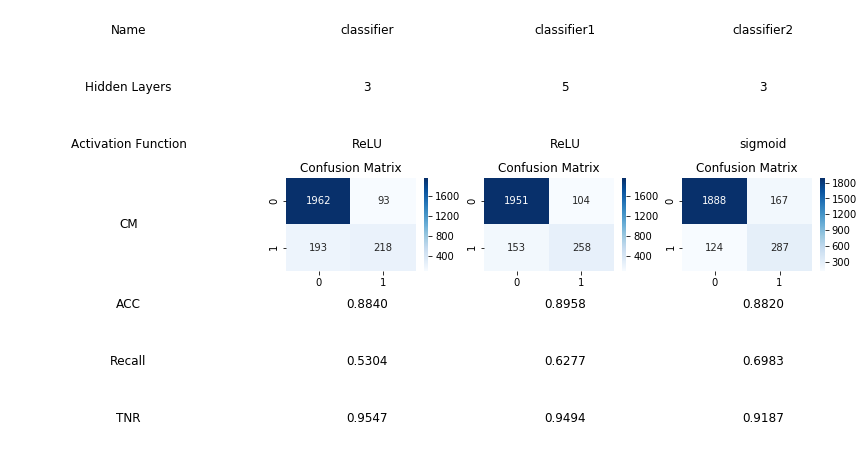
### Train The Models and Compare Performance

Based on the Sequential model in the Keras library, we trained each model for 50 rounds using the training set, not randomly shuffling the data, and used 10% of the training data as a validation set after each training to evaluate the model performance.

After that, we evaluate each model using a test set. We make a confusion matrix of three models.

Considering that our goal is to correctly identify the groups that order and do not order, and to apply different promotional measures to them, we calculated Accuracy, Recall, and Specificity, as shown in Table 13.

Table 13 Comparison of 3 ANN Models



### Prediction result analysis

According to Table 13, Accuracy and TNR of the three models are high, while recall are generally low. This indicates that the model can identify most users who do not place orders, while the predictive power for users who place orders is low. This has much to do with the unbalance of the data set.

Given that we want to use machine learning predictions to target different users with different marketing approaches, we believe that the performance of these three models is mediocre and that further parameter optimization, model replacement, or data set replacement should be considered.

# Experiment 3: Customer clustering analysis of a certain airline based on K-Means algorithm

## Learning Objective

### Machine learning theory: K-means

Through the K-means clustering algorithm, a popular unsupervised learning method for partitioning data into clusters, we aim to understand the general workflow of machine learning tasks, including data preprocessing, feature selection, parameter tuning, and model evaluation.

The process will involve initializing centroids, assigning data points to the nearest cluster center, and iteratively updating the centroid positions based on the mean of the points within each cluster. Through this iterative refinement, we should grasp the basic principle and framework of the K-means algorithm, as well as how the choice of k (the number of clusters) and initial centroid positions can influence the final clustering outcome.

Simultaneously, we intend to learn how to utilize the Python sklearn library in practical applications, specifically focusing on class calls and instantiations related to K-means, such as setting the number of clusters (n\_clusters), using fit or fit\_predict methods for training, and employing silhouette analysis or the elbow method for model evaluation and determining the optimal number of clusters. Additionally, we will explore visualization techniques to interpret the clustering results effectively.

### Experimental objective: Airline Customer Value Analysis

The purpose of this experiment is to classify customers based on airline data, compare the value of different types of customers, and develop sales strategies, in order to achieve personalized services for customers with different values and focus limited resources on high-value customers. Learners can use clustering analysis in this experiment to obtain clustering results and conduct certain analysis on customers of different categories.

## Data Resource

### Data Sources

The data file originates from './air\_data.csv'. This dataset is collected by a certain airline company, with a total of 62988 pieces of data and 44 variables.

### Data Dictionary

Table 14 Experiment 3: Data Dictionary

|  |  |  |
| --- | --- | --- |
| Variable Name | Description | Data Type |
| MEMBER\_NO | Membership number | int64 |
| FFP\_DATE | Enrollment date | object |
| FIRST\_FLIGHT\_DATE | Date of first flight | object |
| GENDER | Gender (‘M’ for male, ‘F’ for female) | object |
| FFP\_TIER | Tier level of frequent flyer program | int64 |
| WORK\_CITY | City where the member works | object |
| WORK\_PROVINCE | Province where the member works | object |
| WORK\_COUNTRY | Country where the member works | object |
| AGE | Age of the member | float64 |
| LOAD\_TIME | Observation window end time | object |
| FLIGHT\_COUNT | Number of flights during observation period | int64 |
| BP\_SUM | Total basic points earned during observation period | int64 |
| EP\_SUM\_YR\_1 | Elite points earned in year 1 | int64 |
| EP\_SUM\_YR\_2 | Elite points earned in year 2 | int64 |
| SUM\_YR\_1 | Total fare paid in year 1 | float64 |
| SUM\_YR\_2 | Total fare paid in year 2 | float64 |
| SEG\_KM\_SUM | Total segment kilometers flown during observation period | int64 |
| WEIGHTED\_SEG\_KM | Weighted total segment kilometers flown during observation period | float64 |
| LAST\_FLIGHT\_DATE | Date of last flight | object |
| AVG\_FLIGHT\_COUNT | Average monthly flight count | float64 |
| AVG\_BP\_SUM | Average monthly basic points earned | float64 |
| BEGIN\_TO\_FIRST | Time from enrollment to first flight | int64 |
| LAST\_TO\_END | Time from last flight to end of observation window | int64 |
| AVG\_INTERVAL | Average interval between consecutive flights | float64 |
| MAX\_INTERVAL | Maximum interval between consecutive flights | int64 |
| ADD\_POINTS\_SUM\_YR\_1 | Other points added in year 1 | int64 |
| ADD\_POINTS\_SUM\_YR\_2 | Other points added in year 2 | int64 |
| EXCHANGE\_COUNT | Number of point exchanges | int64 |
| avg\_discount | Average discount rate | float64 |
| PIY\_Flight\_Count | Number of flights taken in year 1 | int64 |
| LIY\_Flight\_Count | Number of flights taken in year 2 | int64 |
| PIY\_BP\_SUM | Basic points earned in year 1 | int64 |
| LIY\_BP\_SUM | Basic points earned in year 2 | int64 |
| EP\_SUM | Total elite points earned | int64 |
| ADD\_Point\_SUM | Other points added | int64 |
| Eli\_AddPoint\_Sum | Non-flight related points added | int64 |
| LIY\_Eli\_Add\_Points | Non-flight related points added in year 2 | int64 |
| Points\_Sum | Total points earned | int64 |
| LIY\_Points\_Sum | Total points earned in year 2 | int64 |
| Ration\_LIY\_Flight\_Count | Ratio of flights taken in year 2 to total flights | float64 |
| Ration\_PIY\_Flight\_Count | Ratio of flights taken in year 1 to total flights | float64 |
| Ration\_PIY\_BPS | Ratio of basic points earned in year 1 to total basic points | float64 |
| Ration\_LIY\_BPS | Ratio of basic points earned in year 2 to total basic points | float64 |
| Point\_NotFlight | Number of non-flight related point transactions | int64 |

## Algorithm Introduction

### K – means Algorithm

K-means is a popular clustering algorithm for dividing data points into K clusters. Its main goal is to make points within clusters as similar as possible and points between clusters as different as possible.

The K-means algorithm is simple, efficient and suitable for large-scale data sets. However, there are some limitations, such as sensitivity to the initial center of mass and inability to deal with non-spherical clusters.

The basic flow of the k-means algorithm is as follows:

initialize K and centroid of clusters

repeat until the algorithm convergent

Calculate the distance from the data to each central point to form a distance matrix

For each data point, the cluster centroid with the smallest distance is assigned

Calculate SSE= sum of in-cluster errors

Move the centroid, and the new centroid is the intra-group average for each cluster

### RFM Model

RFM model is an important tool and means to measure the current user value and customer potential value. RFM stands for Rencency (last consumption), Frequency (consumption frequency), and Monetary (consumption amount), the first letter combination of the three indicators is as follows:

#### R-value: Last consumption (Recency)

Consumption refers to the time interval between the last and last time a customer spends money in a store. Theoretically, customers with smaller R-value are customers with higher value, that is, they are most likely to respond to the store's repurchases several times. At present, online shopping is convenient, customers have more purchase choices and lower purchase costs, remove the geographical constraints, customers are very easy to lose, so CRM operators want to improve the buyback rate and retention rate, need to be alert to R value.

#### F-value: Consumption Frequency

Consumption frequency is the number of purchases a customer makes in a fixed period of time (usually 1 year). However, if the actual store in the actual operation is due to the width of the category, such as selling 3C products, durable goods and so on, even loyal fans are difficult to buy several times in a year. Therefore, when the general store operates the RFM model, it will remove the time range of the F-value and replace it with the cumulative number of purchases.

#### M value: Monetary

M-value is the most difficult to use in RFM model compared to R-value and F-value, but the most valuable index. The familiar "80/20 rule" (also known as the "Pareto rule") once explained that 80% of a company's revenue comes from 20% of its users, and in theory, the M value and the F value are the same, both with a time horizon, referring to the amount of money spent over a period of time (usually 1 year), but the M value has a relatively weak role in customer segmentation

### LRMFC Model

Generally, the most extensive model we use to identify customer value is the RFM model, but different passengers with the same amount of consumption have different values to airlines. For example, passengers who buy long routes and low class and passengers who buy short routes and high class have the same amount of consumption, and their values are indeed different, obviously the latter is more valuable. Therefore, only RFM three indicators may not be appropriate, we need to add new indicators on the basis of RFM model to analyze airline customers

From this, we construct the LRFMC model, in which:

L*:* The time between the time of membership and the end of the observation window

R: Time between the end of the observation window and the customer's last ride (number of months)

F: The number of times the customer has flown the company aircraft during the observation window

M: Frequent flyer miles accumulated by the customer during the observation window

C: The average of the discount factor corresponding to the passenger's travel space in the observation window

## Experimental process and Result Analysis

### Data Standardization

Due to the presence of null values in some variables, we first performed data cleaning to eliminate them. After data cleaning, the data volume has been reduced from 62988 to 62044. For the subsequent clustering modeling, we need to standardize the data. The standardized data is show as Table 15.

Table 15 Standardized Data



### Establish K-Means clustering model

In order to obtain the optimal clustering, we need to calculate the contour coefficients at different k values for comparison, as shown in Figure 10.

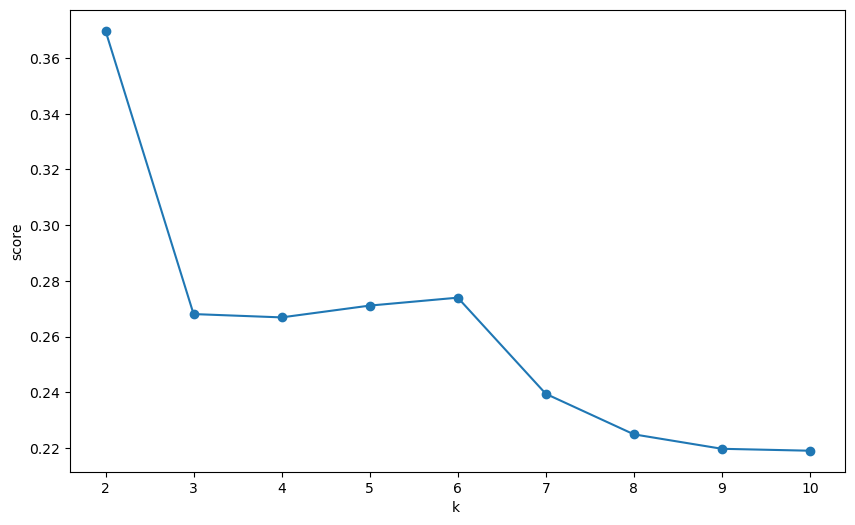


Figure 10 Relationship diagram between contour coefficient and k value

The silhouette coefficient initially drops sharply, then slightly recovers at k=6, and continues to decline until k=10. Specifically, when k increases from 2 to 3, the silhouette coefficient decreases significantly, indicating a deterioration in clustering quality. Atk=6, there is a small increase in the silhouette coefficient, suggesting an improvement in cluster structure at this specific k value. However, from k=7 onwards, the silhouette coefficient continues to decline, indicating that the clustering quality worsens as the number of clusters further increases.

### Optimal k-value selection and visualization

After comparison, we found that the optimal k value is 2, and the clustering center is: [[ 0.22004536 -0.76242259 1.79802489 1.7461081 0.32968294]

[-0.03993282 0.13836096 -0.32629731 -0.31687569 -0.05982935]]

The clustering categories are: [0 0 0 ... 1 1 1]

Based on this, we will draw a visualization of the results of customer value clustering. We have drawn the following graph as Figure 11.

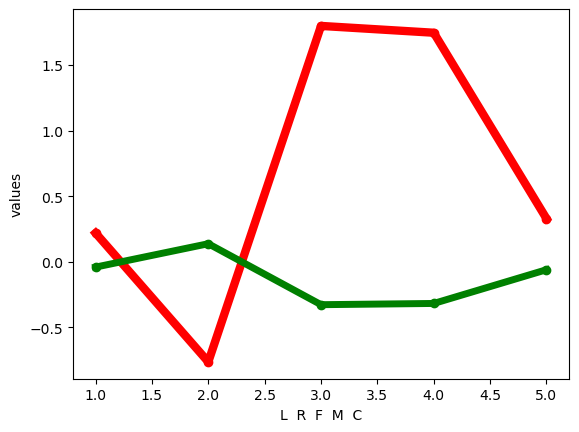


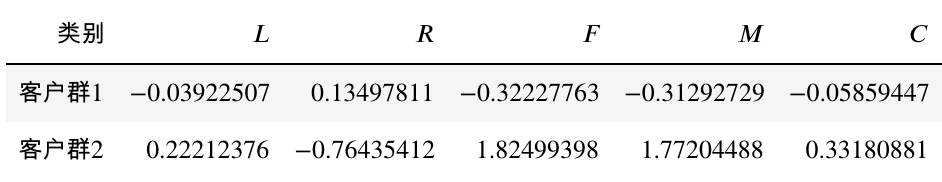
Figure 11 Customer Value Clustering Line Chart

It should be noted that every time we run the k-means clustering model (without setting the number of states), the classes we obtain will differ and the cluster numbers will change accordingly, but the clustering center points will not change significantly.

### Customer Value analysis

We then conducted a customer value analysis, and the following table corresponds one-to-one with the data in the line graph above. The different colors in the graph represent different customer groups and clusters. The horizontal axis corresponds to the columns in the table below. The table is shown in Table 16.

Table 16 Customer Value Analysis



Because there may be differences in the order of clustering each category, we conduct value analysis on customer groups based on clustering centers

Because our focus is on the indicator L, F，M， From the visualization chart, we can obtain the following results:

1. Customer group 1 [red] has the highest value at feature R and a smaller value at feature F, indicating that customer group 1 has not taken a flight for a long time and is a low value customer group with a short membership time

2. Customer group 2 [green] has the highest values on features F and M, and the lowest value on feature R, indicating that members of customer group 4 frequently fly and have recent flight records.

# Experiment 4: Product Recommendation Based on Association Rules

## Learning Objective

### Data mining: slicing, dicing, and visualization

In order to initially explore the interesting rules in the dataset, we should first perform descriptive statistics on the data, slice and dice the data by month, season, time period, and so on, and get some insights from the visualizations.

### Machine Learning Theory: Association Rules

In order to get more scientific and more confident rules, we should use the steps of association rules to generate rules and evaluate them. We should learn the basic steps of association rules in the experiment, as well as evaluation metrics (support, confidence, lift).

## Data Source

Table 17 Experiment 4:Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Feature Name | Data Type | Description | Form or Possible Values |
| Transaction | int64 | Transaction ID, same ID indicating that different items are purchased together in the same transaction. |  |
| Item | object | Name of the product purchased | 94 items in total |
| date\_time | object | Transaction day and time | “Day-Month-Year Hour:Minute” |
| period\_day | object | Time of day when the transaction occurred | “morning”, “afternoon”, “evening”, “night” |
| weekday\_weekend | object | Whether it is a weekend or not. | 0 = not,  1 = yes |

## Introduction of Apriori Algorithm

Association rules are mainly used to discover interesting relationships between items from large-scale data sets, which can be expressed as frequent item sets, associations, correlations, causal structures, etc.

### Three Important Indicators of Association Rules

#### Support

#### Confidence

#### Lift

### Apriori Principle (Antimonotonicity)

All subsets of a frequent itemset must be frequent(Apriori property).

If an itemset is infrequent, all its supersets will be infrequent.

### Apriori Procedure

Find all C1(candidate 1-itemset)

Using min support, select L1(Large 1-itemset)

For k=2,3,......

Using the join rule, expand Ck-1 to get Ck

Use Antimonotonicity to check Ck, remove Ck if Ck or any subset of Ck is not a frequent itemset

Get Lk

if there are no elements in Lk, break

Use all L to generate rules

### Generate all Association Rules

#### Generation rules

For each frequent itemset L, generate all nonempty subsets of L.

For each nonempty subset s of L, generate a rule s→ (G-s).

#### Pruning

Calculate the confidence for each rule.

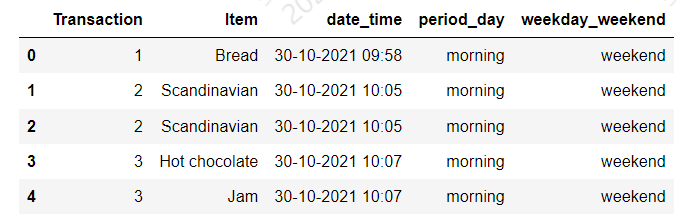
Remove a rule if its confidence does not meet min-Conf.

## Experimental Process and Result Analysis

### Data Preprocessing: Data Transformation of Time Type

Using .head() function, we first took a rough look at the data, as shown in Table 18.

Table 18 Raw Data Head

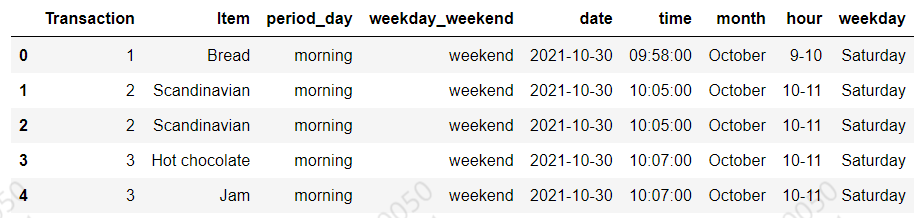


Considering that the format of the date\_time attribute in the original data set is not normal, we first convert this attribute using pd.to\_datetime.

In order to extract more information, we refine the processed datetime and extract 5 different attributes of “date”, “time”, “month”, “hour” and “weekday”.

The converted data is shown in Table 19.

Table 19 Data Head after Coding



### Data exploration: Descriptive statistics and visualization

#### The most popular products

We sorted the purchase frequency of individual products, presenting the top 20 most popular products and their purchase frequency in the form of a bar chart, as shown in Figure 12.

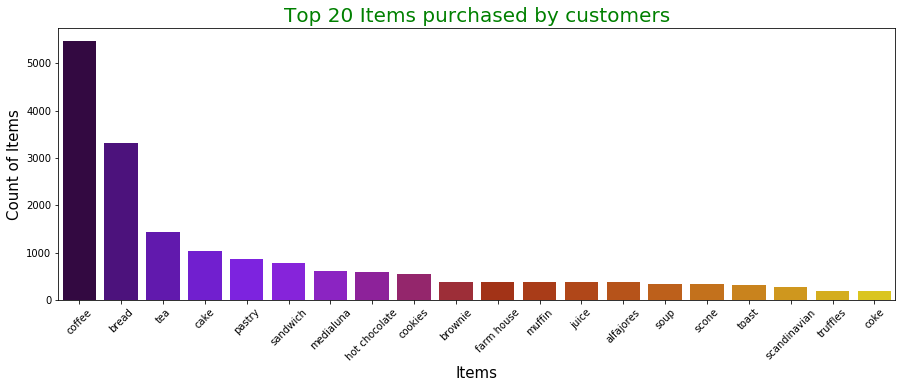


Figure 12 Top 20 ltems purchased

#### Number of Transactions Per Month

We made slices according to the month and counted the number of products sold in each month, as shown in Figure 13.

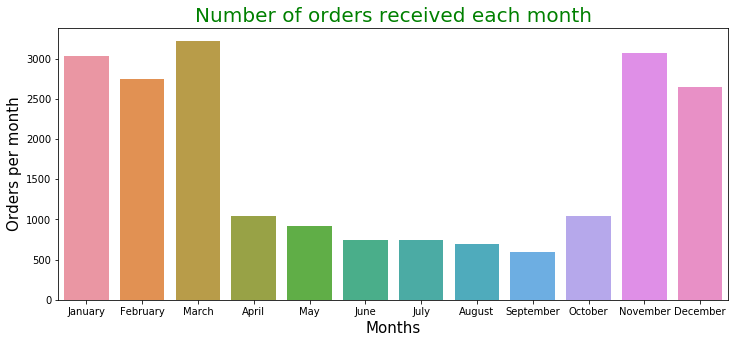


Figure 13 Number of orders received each month

From Figure 13, it is interesting to see that the majority of transactions take place between November and March.

#### Number of Transactions Per Weekday

We made slices according to weekday and counted the number of products sold on different days of the week, as shown in Figure 14



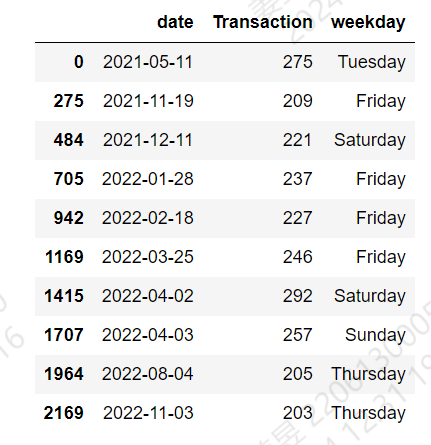
Figure 14 Number of orders received each day.

According to Figure 14, people consume slightly more on weekends than in midweek.

#### Hot Day

We counted the days when sales exceeded 200, as shown in Table 20.

Table 20 Daily Transaction over 200



Daily trading exceeds 200 mostly in winter, and all on Thursday, Friday, Saturday and Sunday.

#### Purchasing Time and Period

We conducted and sliced period\_day and hour to calculate the sales in different periods of a day, as shown in Figure 15 and Figure 16.



Figure 15 Count of orders received each hour



Figure 16 Count of orders received each periodof a day

It can be seen that trading is concentrated between 8 a.m. and 5 p.m.

In general, people are more likely to spend money in the morning and afternoon.

### Dice: Time and Item

We were interested in the impact of different times and seasons on best-selling products, so we conducted a multi-dimensional slice analysis.

#### Period-Item Dicing

First, we diced the data base on Period\_day and item and found the best-selling products in 4 periods, as shown in Figure 17.

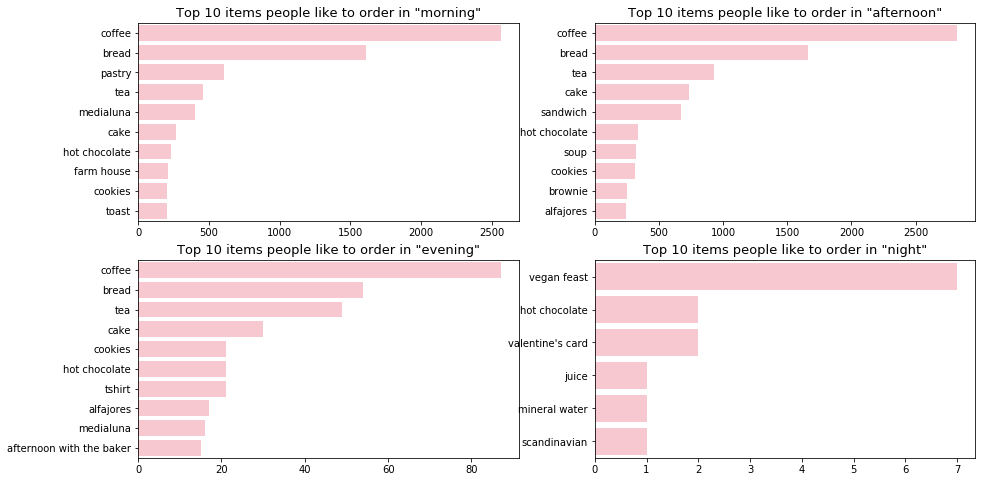


Figure 17 Top 10 most sold products in each period

As can be seen from Figure 17, the sales volume of goods at night is very low, so we can ignore it. In the other three periods, the sales of coffee and bread remained in the top two, and the sales of cake and tea were also relatively high.

#### Season-Item

We have also diced the transaction counts according to season and item, and calculated the best-selling products of the four seasons, as shown in Figure 18.

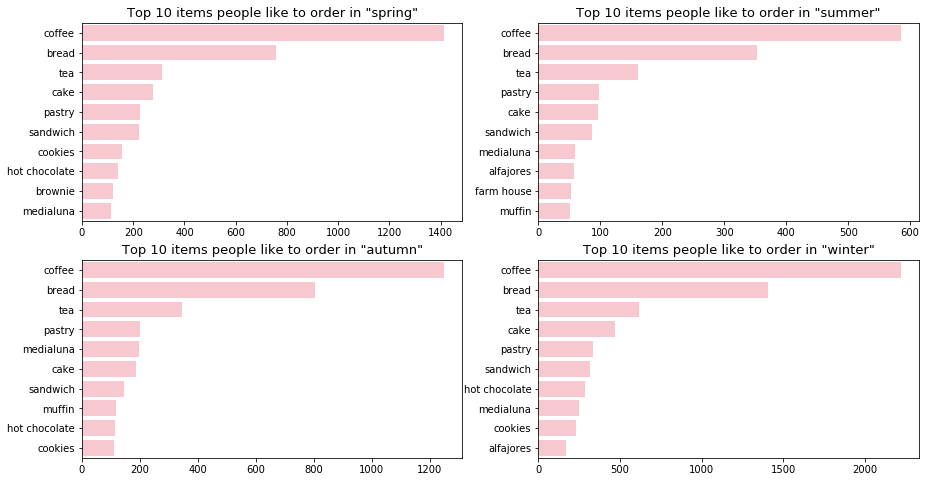


Figure 18 Top 10 most sold products in each season

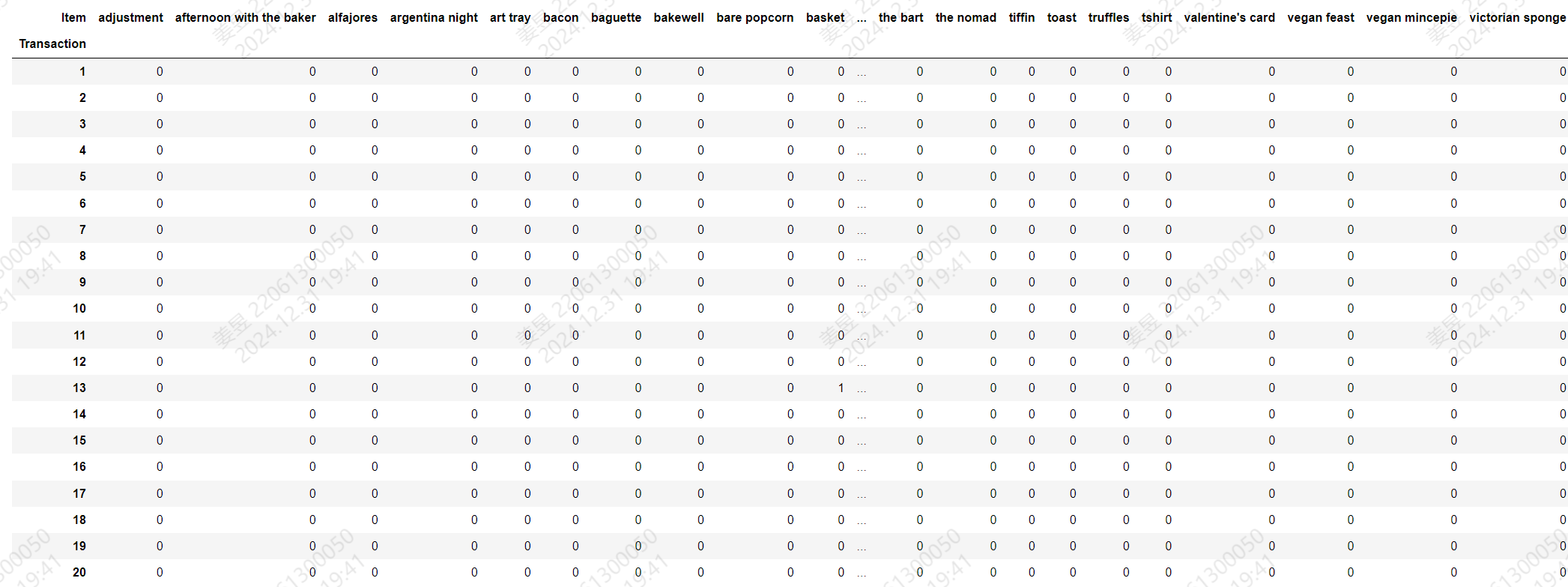
As can be seen from Figure 18, no matter what season, coffee,bread and tea are always people's favorite commodities, which is also in line with common sense.

### Generation of Association Rule

#### Construct a Transaction Table

We first build the transaction table, where each row represents a transaction, for a total of 9,465 rows, indicating that 9,465 transactions were counted. Each column is a product, where 1 means purchased in that transaction and 0 means not purchased. The portion of the transaction table is shown in Table 21.

Table 21 Transaction Table



#### Query Frequent Itemsets

We use the apriori algorithm to find all frequent item sets and set min\_support = 0.01. All frequent item sets are shown in Table 22.

Table 22 Frequent Items

|  |  |  |  |
| --- | --- | --- | --- |
| support | itemsets | support | itemsets |
| 0.0363 | alfajores | 0.0197 | alfajores', 'coffee |
| 0.0161 | baguette | 0.0108 | brownie', 'bread |
| 0.3272 | bread | 0.0233 | bread', 'cake |
| 0.0400 | brownie | 0.0900 | coffee', 'bread |
| 0.1039 | cake | 0.0145 | cookies', 'bread |
| 0.0130 | chicken stew | 0.0134 | hot chocolate', 'bread |
| 0.4784 | coffee | 0.0169 | medialuna', 'bread |
| 0.0194 | coke | 0.0292 | bread', 'pastry |
| 0.0544 | cookies | 0.0170 | sandwich', 'bread |
| 0.0392 | farm house | 0.0281 | tea', 'bread |
| 0.0150 | fudge | 0.0197 | coffee', 'brownie |
| 0.0106 | hearty & seasonal | 0.0547 | coffee', 'cake |
| 0.0583 | hot chocolate | 0.0114 | hot chocolate', 'cake |
| 0.0150 | jam | 0.0238 | tea', 'cake |
| 0.0132 | jammie dodgers | 0.0282 | cookies', 'coffee |
| 0.0386 | juice | 0.0296 | hot chocolate', 'coffee |
| 0.0618 | medialuna | 0.0206 | juice', 'coffee |
| 0.0142 | mineral water | 0.0352 | coffee', 'medialuna |
| 0.0385 | muffin | 0.0188 | coffee', 'muffin |
| 0.0861 | pastry | 0.0475 | coffee', 'pastry |
| 0.0105 | salad | 0.0382 | sandwich', 'coffee |
| 0.0718 | sandwich | 0.0181 | scone', 'coffee |
| 0.0291 | scandinavian | 0.0158 | coffee', 'soup |
| 0.0345 | scone | 0.0109 | spanish brunch', 'coffee |
| 0.0344 | soup | 0.0499 | tea', 'coffee |
| 0.0182 | spanish brunch | 0.0237 | coffee', 'toast |
| 0.1426 | tea | 0.0144 | sandwich', 'tea |
| 0.0154 | tiffin | 0.0100 | coffee', 'bread', 'cake |
| 0.0336 | toast | 0.0112 | coffee', 'bread', 'pastry |
| 0.0203 | truffles | 0.0100 | tea', 'coffee', 'cake |
| 0.0104 | alfajores', 'bread |  |  |

According to Table 22, there are 61 frequent item sets, and the maximum length of the frequent item set is 4. Although there are 94 different products, there is not a large portfolio of products that are purchased multiple times at the same time.

#### Generate Association Rules

Using mlxtend.frequent\_patterns. association\_rules, we can directly obtain association rules and display key indicators such as support, confidence, and promotion.

We set lift as the criterion for association rules, and any rule with a lift greater than 1 will be retained. A total of 42 association rules are formed, as shown in Table 22.

Table 23 Association Rules

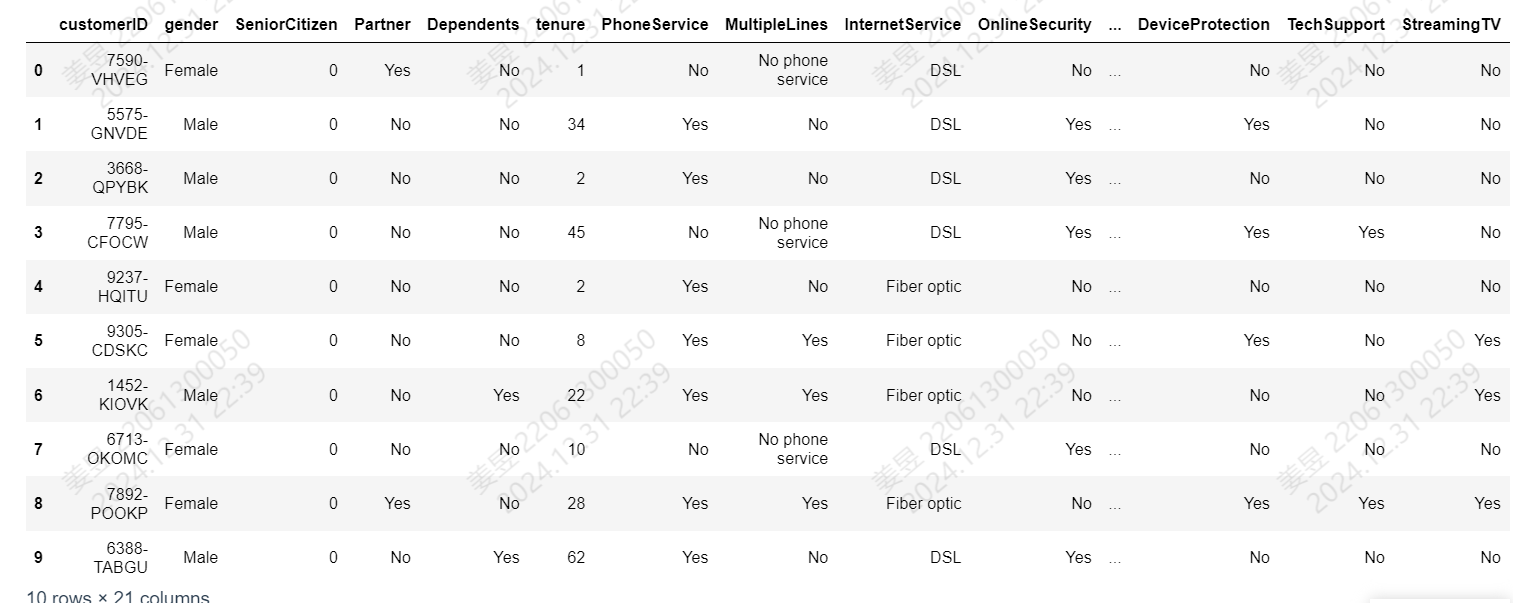
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **antecedent** | **consequent** | **Ante - sup** | **Conse - sup** | **sup** | **conf** | **lift** | **leverage** | **conviction** |
| toast | coffee | 0.0336 | 0.4784 | 0.0237 | 0.7044 | 1.4724 | 0.0076 | 1.7646 |
| spanish brunch | coffee | 0.0182 | 0.4784 | 0.0109 | 0.5988 | 1.2518 | 0.0022 | 1.3002 |
| medialuna | coffee | 0.0618 | 0.4784 | 0.0352 | 0.5692 | 1.1899 | 0.0056 | 1.2109 |
| pastry | coffee | 0.0861 | 0.4784 | 0.0475 | 0.5521 | 1.1542 | 0.0064 | 1.1647 |
| alfajores | coffee | 0.0363 | 0.4784 | 0.0197 | 0.5407 | 1.1302 | 0.0023 | 1.1356 |
| juice | coffee | 0.0386 | 0.4784 | 0.0206 | 0.5342 | 1.1167 | 0.0022 | 1.1199 |
| sandwich | coffee | 0.0718 | 0.4784 | 0.0382 | 0.5324 | 1.1128 | 0.0039 | 1.1154 |
| cake | coffee | 0.1039 | 0.4784 | 0.0547 | 0.5270 | 1.1015 | 0.0050 | 1.1027 |
| scone | coffee | 0.0345 | 0.4784 | 0.0181 | 0.5229 | 1.0931 | 0.0015 | 1.0934 |
| cookies | coffee | 0.0544 | 0.4784 | 0.0282 | 0.5184 | 1.0837 | 0.0022 | 1.0832 |
| hot chocolate | coffee | 0.0583 |  | 0.0296 | 0.5072 | 1.0603 | 0.0017 | 1.0586 |
| brownie | coffee | 0.0400 | 0.4784 | 0.0197 | 0.4908 | 1.0259 | 0.0005 | 1.0243 |
| muffin | coffee | 0.0385 | 0.4784 | 0.0188 | 0.4890 | 1.0222 | 0.0004 | 1.0208 |
| pastry | bread | 0.0861 | 0.3272 | 0.0292 | 0.3387 | 1.0350 | 0.0010 | 1.0173 |
| cake | tea | 0.1039 | 0.1426 | 0.0238 | 0.2289 | 1.6048 | 0.0090 | 1.1119 |
| tea,  coffee | cake | 0.0499 | 0.1039 | 0.0100 | 0.2013 | 1.9380 | 0.0049 | 1.1220 |
| sandwich | tea | 0.0718 | 0.1426 | 0.0144 | 0.2000 | 1.4022 | 0.0041 | 1.0717 |
| hot chocolate | cake | 0.0583 | 0.1039 | 0.0114 | 0.1957 | 1.8839 | 0.0054 | 1.1141 |
| coffee,cake | tea | 0.0547 | 0.1426 | 0.0100 | 0.1834 | 1.2858 | 0.0022 | 1.0499 |
| tea | cake | 0.1426 | 0.1039 | 0.0238 | 0.1667 | 1.6048 | 0.0090 | 1.0754 |
| pastry | coffee,bread | 0.0861 | 0.0900 | 0.0112 | 0.1301 | 1.4449 | 0.0034 | 1.0460 |
| coffee,bread | pastry | 0.0900 | 0.0861 | 0.0112 | 0.1244 | 1.4449 | 0.0034 | 1.0437 |
| coffee | cake | 0.4784 | 0.1039 | 0.0547 | 0.1144 | 1.1015 | 0.0050 | 1.0119 |
| coffee,bread | cake | 0.0900 | 0.1039 | 0.0100 | 0.1115 | 1.0736 | 0.0007 | 1.0086 |
| cake | hot chocolate | 0.1039 | 0.0583 | 0.0114 | 0.1099 | 1.8839 | 0.0054 | 1.0579 |
| tea | sandwich | 0.1426 | 0.0718 | 0.0144 | 0.1007 | 1.4022 | 0.0041 | 1.0321 |
| coffee | pastry | 0.4784 | 0.0861 | 0.0475 | 0.0994 | 1.1542 | 0.0064 | 1.0147 |
| cake | coffee,bread | 0.1039 | 0.0900 | 0.0100 | 0.0966 | 1.0736 | 0.0007 | 1.0073 |
| cake | tea,coffee | 0.1039 | 0.0499 | 0.0100 | 0.0966 | 1.9380 | 0.0049 | 1.0518 |
| bread | pastry | 0.3272 | 0.0861 | 0.0292 | 0.0891 | 1.0350 | 0.0010 | 1.0033 |
| coffee | sandwich | 0.4784 | 0.0718 | 0.0382 | 0.0799 | 1.1128 | 0.0039 | 1.0088 |
| coffee | medialuna | 0.4784 | 0.0618 | 0.0352 | 0.0735 | 1.1899 | 0.0056 | 1.0127 |
| tea | coffee,cake | 0.1426 | 0.0547 | 0.0100 | 0.0704 | 1.2858 | 0.0022 | 1.0168 |
| coffee | hot chocolate | 0.4784 | 0.0583 | 0.0296 | 0.0618 | 1.0603 | 0.0017 | 1.0037 |
| coffee | cookies | 0.4784 | 0.0544 | 0.0282 | 0.0590 | 1.0837 | 0.0022 | 1.0048 |
| coffee | toast | 0.4784 | 0.0336 | 0.0237 | 0.0495 | 1.4724 | 0.0076 | 1.0167 |
| coffee | juice | 0.4784 | 0.0386 | 0.0206 | 0.0431 | 1.1167 | 0.0022 | 1.0047 |
| coffee | brownie | 0.4784 | 0.0400 | 0.0197 | 0.0411 | 1.0259 | 0.0005 | 1.0011 |
| coffee | alfajores | 0.4784 | 0.0363 | 0.0197 | 0.0411 | 1.1302 | 0.0023 | 1.0049 |
| coffee | muffin | 0.4784 | 0.0385 | 0.0188 | 0.0393 | 1.0222 | 0.0004 | 1.0009 |
| coffee | scone | 0.4784 | 0.0345 | 0.0181 | 0.0378 | 1.0931 | 0.0015 | 1.0033 |
| coffee | spanish brunch | 0.4784 | 0.0182 | 0.0109 | 0.0227 | 1.2518 | 0.0022 | 1.0047 |

Based on the confidence ranking obtained in Table 23, we can see that the top three commodity combinations are toast-coffee,spanish brunch-coffee and medialuna-coffee. We can do bundled sales promotion according to this, and put related goods together to develop corresponding sales packages to promote sales and increase profits.

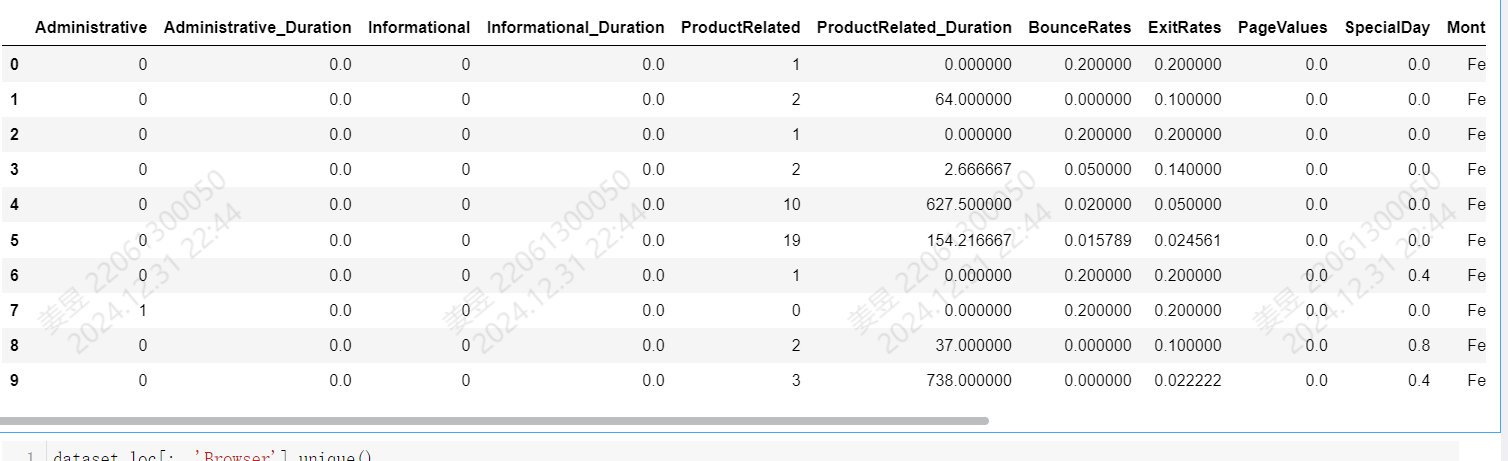
# Appendix

## Raw data extract

### Experiment 1



### Experiment 2



### Experiment 3



### Experiment 4