Lab2 - Logistic regression (Titanic dataset)

WEN Yue
CHEN MengYu

1. Before logistic regression

(1) About this dataset

This is a dataset about some basic information of the passengers who was on Titanic.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

(2) Basic analyze

(a) By using the command ***data.info()***, we can see that there are some nulls in this dataset, so we need to clean this dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

Before cleaning, we can do some simple analyze first.

(b) As can be seen from the results of the table, The average survival rate was 0.383838, indicating a large rife, and Pclass's average was 2.3, indicating that passengers in third-class cabins were the most expensive, and the average age was 29.7 years old, which showed that many adults had younger children, resulting in a smaller average age.

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

(c) Data cleaning

- To facilitate analysis, we have added a new column, Family
  SibSp+ Parch=0 => Family = 0
  SibSp+ Parch>0 and SibSp+ Parch<4 => Family = 1
  SibSp+ Parch>4 => Family = 2
- Filling missing age attributes with **RandomForestClassifier**
- Remove useless columns

	PassengerId	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked	Family
0	1	0	3	male	22.0	-0.533749	No	S	1
1	2	1	1	female	38.0	0.864760	Yes	C	1
2	3	1	3	female	26.0	-0.519007	No	S	0
3	4	1	1	female	35.0	0.467631	Yes	S	1
4	5	0	3	male	35.0	-0.516277	No	S	0

- In order to better perform logistic regression, we factorize some variables.

Cabin_No	Cabin_Yes	Embarked_C	Embarked_Q	Embarked_S	Sex_female	Sex_male	Pclass_1	Pclass_2	Pclass_3	Family_0	Family_1	Family_2
1	0	0	0	1	0	1	0	0	1	0	1	0
0	1	1	0	0	1	0	1	0	0	0	1	0
1	0	0	0	1	1	0	0	0	1	1	0	0
0	1	0	0	1	1	0	1	0	0	0	1	0
1	0	0	0	1	0	1	0	0	1	1	0	0

- The range of data values for the two attributes Age and Fare is too large, which will adversely affect the convergence of logistic regression. The solution is to standardize it.

2. Scikit Learn

(1) At beginning, we split the dataset into two parts, train and test. After cleaning the data from these two parts, we can use the logistic regression model.

(2) After training, we can use the model to make a prediction.

	PassengerId	Survived
0	625	0
1	626	0
2	627	0
3	628	1
4	629	0

(3) We can calculate the accuracy of the prediction: 80.524%

3. Custom method

(1) After data cleaning, we can get our new dataset.

	Survived	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	0	3	22.0	1	0	7.2500	1	0	1
1	1	1	38.0	1	0	71.2833	0	0	0
2	1	3	26.0	0	0	7.9250	0	0	1
3	1	1	35.0	1	0	53.1000	0	0	1
4	0	3	35.0	0	0	8.0500	1	0	1

(2) Define sigmoid function.

$$f(z) = \frac{1}{1 + e^{-z}}$$

```
def sigmoid(z):
    return float(1 / float((1 + math.exp(-1 * z))))
```

(3) Define hypothesis function.

```
def hypothesis(theta,x):
    z = 0
    for i in range(len(theta)):
        z += theta[i] * x[i]s
    return sigmoid(z)
```

(4) Define cost function.

$$L(h_\theta(x), y) = -y \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))$$

```
def cost_function(X,y,theta,m):
    sumError = 0
    for i in range(m):
        error = 0
        x_i = X[i]
        y_i = y[i]
        h_i = hypothesis(theta,x_i)
        error = y_i * math.log(h_i) + (1 - y_i) * math.log(1 - h_i)
        sumError += error
    const = -1/m
    cost = const * sumError
    return cost
```

(5) Define cfd

```
def cfd(X,y,theta,j,m):
    sumError = 0
    for i in range(m):
        x_i = X[i]
        x_ij = x_i[j]
        h_i = hypothesis(theta, x_i)
        error = (h_i - y[i]) * x_ij
        sumError += error
    const = 1/m
    cost = const * sumError
    return cost
```

(6) Define gradient_descent.

```
def gradient_descent(X,y,theta,alpha,m):
    opt_theta = []
    for j in range(len(theta)):
        cost = cfd(X,y,theta,j,m)
        updated_theta = theta[j] - (alpha * cost)
        opt_theta.append(updated_theta)
    return opt_theta
```

(7) Define gradient_descent

```
def Logistic_Regression(X, y, alpha, theta, iterations):
    m = len(y)
    for i in range(iterations):
        opt_theta = gradient_descent(X,y,theta,alpha,m)
        theta = opt_theta
    return theta
```

(8) Now, we can get the coefficients of the logistic regression model

```
[2.782986228544027,
 0.3585247001191568,
 -6.835135343154106,
 -2.054051750981707,
 0.4866005435831 35,
 -29.011275367203528,
 -1.8211726628704537,
 -2.895795707817223]
```

(9) We can calculate the accuracy of the prediction: 71.067%.

4. Conclusions

Through the above description, the prediction accuracy of custom logistic regression is lower than the accuracy using sklearn.Prior to custom logistic regression, we did not perform the same data cleanup on the dataset, so the accuracy rate was lower.