**源代码**

# **调库及准备工作**

In [20]:

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

*#import missingno as msno*

​

**%**matplotlib inline

​

**from** IPython.core.interactiveshell **import** InteractiveShell

InteractiveShell.ast\_node\_interactivity **=** "all" *# 一个cell显示多输出*

​

*#pd.set\_option('display.max\_rows', 100) # 设置最大显示100行*

*#pd.set\_option('display.max\_columns', 100) # 设置最大显示100列*

# **调用数据及查看数据**

In [21]:

train\_data **=** pd.read\_csv("d:/工作室/QG/train and test2/train.csv")

*#train1 = pd.read\_csv("./train.csv")*

*#test1 = pd.read\_csv("d:/工作室/QG/train and test1/test1.csv")*

train\_data

*#test1*

Out[21]:

|  | **id\_num** | **program\_type** | **program\_id** | **program\_duration** | **test\_id** | **test\_type** | **difficulty\_level** | **trainee\_id** | **gender** | **education** | **city\_tier** | **age** | **total\_programs\_enrolled** | **is\_handicapped** | **trainee\_engagement\_rating** | **is\_pass** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 9389\_150 | Y | Y\_1 | 136.0 | 150.0 | offline | intermediate | 9389.0 | M | Matriculation | 3.0 | 24.0 | 5.0 | N | 1.0 | 0 |
| **1** | 16523\_44 | T | T\_1 | 131.0 | 44.0 | offline | easy | 16523.0 | F | High School Diploma | 4.0 | 26.0 | 2.0 | N | 3.0 | 1 |
| **2** | 13987\_178 | Z | Z\_2 | 120.0 | 178.0 | online | easy | 13987.0 | M | Matriculation | 1.0 | 40.0 | 1.0 | N | 2.0 | 1 |
| **3** | 13158\_32 | T | T\_2 | 117.0 | 32.0 | offline | easy | 13158.0 | F | Matriculation | 3.0 | NaN | 4.0 | N | 1.0 | 1 |
| **4** | 10591\_84 | V | V\_3 | 131.0 | 84.0 | offline | intermediate | 10591.0 | F | High School Diploma | 1.0 | 42.0 | 2.0 | N | 4.0 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **49993** | 4379\_115 | Y | Y\_4 | 121.0 | 115.0 | offline | hard | 4379.0 | F | High School Diploma | 2.0 | NaN | 5.0 | N | 4.0 | 1 |
| **49994** | 3809\_58 | U | U\_1 | NaN | 58.0 | offline | intermediate | 3809.0 | F | High School Diploma | 3.0 | 30.0 | 2.0 | N | 3.0 | 1 |
| **49995** | 18109\_42 | T | T\_2 | 117.0 | 42.0 | online | easy | 18109.0 | F | High School Diploma | 2.0 | 41.0 | 2.0 | N | 1.0 | 1 |
| **49996** | 6384\_113 | Y | Y\_4 | 121.0 | 113.0 | offline | intermediate | 6384.0 | M | Matriculation | 3.0 | NaN | 2.0 | N | 5.0 | 1 |
| **49997** | 18598\_111 | X | X\_1 | 134.0 | 111.0 | offline | hard | 18598.0 | M | Matriculation | 2.0 | 29.0 | 5.0 | N | NaN | 1 |

49998 rows × 16 columns

In [22]:

train\_data.head(5) *#显示前5行数据*

train\_data.tail(5) *#显示后5行*

train\_data.columns *#查看列名*

train\_data.info() *#查看各字段的信息*

train\_data.shape *#查看数据集行列分布，几行几列*

train\_data.describe() *#查看数据的大体情况*

Out[22]:

|  | **id\_num** | **program\_type** | **program\_id** | **program\_duration** | **test\_id** | **test\_type** | **difficulty\_level** | **trainee\_id** | **gender** | **education** | **city\_tier** | **age** | **total\_programs\_enrolled** | **is\_handicapped** | **trainee\_engagement\_rating** | **is\_pass** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 9389\_150 | Y | Y\_1 | 136.0 | 150.0 | offline | intermediate | 9389.0 | M | Matriculation | 3.0 | 24.0 | 5.0 | N | 1.0 | 0 |
| **1** | 16523\_44 | T | T\_1 | 131.0 | 44.0 | offline | easy | 16523.0 | F | High School Diploma | 4.0 | 26.0 | 2.0 | N | 3.0 | 1 |
| **2** | 13987\_178 | Z | Z\_2 | 120.0 | 178.0 | online | easy | 13987.0 | M | Matriculation | 1.0 | 40.0 | 1.0 | N | 2.0 | 1 |
| **3** | 13158\_32 | T | T\_2 | 117.0 | 32.0 | offline | easy | 13158.0 | F | Matriculation | 3.0 | NaN | 4.0 | N | 1.0 | 1 |
| **4** | 10591\_84 | V | V\_3 | 131.0 | 84.0 | offline | intermediate | 10591.0 | F | High School Diploma | 1.0 | 42.0 | 2.0 | N | 4.0 | 1 |

Out[22]:

|  | **id\_num** | **program\_type** | **program\_id** | **program\_duration** | **test\_id** | **test\_type** | **difficulty\_level** | **trainee\_id** | **gender** | **education** | **city\_tier** | **age** | **total\_programs\_enrolled** | **is\_handicapped** | **trainee\_engagement\_rating** | **is\_pass** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **49993** | 4379\_115 | Y | Y\_4 | 121.0 | 115.0 | offline | hard | 4379.0 | F | High School Diploma | 2.0 | NaN | 5.0 | N | 4.0 | 1 |
| **49994** | 3809\_58 | U | U\_1 | NaN | 58.0 | offline | intermediate | 3809.0 | F | High School Diploma | 3.0 | 30.0 | 2.0 | N | 3.0 | 1 |
| **49995** | 18109\_42 | T | T\_2 | 117.0 | 42.0 | online | easy | 18109.0 | F | High School Diploma | 2.0 | 41.0 | 2.0 | N | 1.0 | 1 |
| **49996** | 6384\_113 | Y | Y\_4 | 121.0 | 113.0 | offline | intermediate | 6384.0 | M | Matriculation | 3.0 | NaN | 2.0 | N | 5.0 | 1 |
| **49997** | 18598\_111 | X | X\_1 | 134.0 | 111.0 | offline | hard | 18598.0 | M | Matriculation | 2.0 | 29.0 | 5.0 | N | NaN | 1 |

Out[22]:

Index(['id\_num', 'program\_type', 'program\_id', 'program\_duration', 'test\_id',

'test\_type', 'difficulty\_level', 'trainee\_id', 'gender', 'education',

'city\_tier', 'age', 'total\_programs\_enrolled', 'is\_handicapped',

'trainee\_engagement\_rating', 'is\_pass'],

dtype='object')

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 49998 entries, 0 to 49997

Data columns (total 16 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 id\_num 49998 non-null object

1 program\_type 49267 non-null object

2 program\_id 49299 non-null object

3 program\_duration 49323 non-null float64

4 test\_id 49273 non-null float64

5 test\_type 49296 non-null object

6 difficulty\_level 49295 non-null object

7 trainee\_id 49259 non-null float64

8 gender 49291 non-null object

9 education 49296 non-null object

10 city\_tier 49298 non-null float64

11 age 30619 non-null float64

12 total\_programs\_enrolled 49306 non-null float64

13 is\_handicapped 49280 non-null object

14 trainee\_engagement\_rating 49226 non-null float64

15 is\_pass 49998 non-null int64

dtypes: float64(7), int64(1), object(8)

memory usage: 6.1+ MB

Out[22]:

(49998, 16)

Out[22]:

|  | **program\_duration** | **test\_id** | **trainee\_id** | **city\_tier** | **age** | **total\_programs\_enrolled** | **trainee\_engagement\_rating** | **is\_pass** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 49323.000000 | 49273.000000 | 49259.000000 | 49298.000000 | 30619.000000 | 49306.000000 | 49226.000000 | 49998.000000 |
| **mean** | 128.229366 | 91.414345 | 9863.493128 | 2.249097 | 36.514256 | 2.583114 | 2.397818 | 0.696288 |
| **std** | 6.889967 | 51.307852 | 5716.490640 | 1.010896 | 9.045487 | 1.239399 | 1.326378 | 0.459864 |
| **min** | 117.000000 | 0.000000 | 1.000000 | 1.000000 | 17.000000 | 1.000000 | 1.000000 | 0.000000 |
| **25%** | 121.000000 | 45.000000 | 5051.500000 | 1.000000 | 28.000000 | 2.000000 | 1.000000 | 0.000000 |
| **50%** | 131.000000 | 91.000000 | 9665.000000 | 2.000000 | 40.000000 | 2.000000 | 2.000000 | 1.000000 |
| **75%** | 134.000000 | 135.000000 | 14618.000000 | 3.000000 | 45.000000 | 3.000000 | 4.000000 | 1.000000 |
| **max** | 136.000000 | 187.000000 | 20097.000000 | 4.000000 | 63.000000 | 14.000000 | 5.000000 | 1.000000 |

In [23]:

total **=** train\_data.isnull().sum().sort\_values(ascending**=False**)

print(total)

age 19379

trainee\_engagement\_rating 772

trainee\_id 739

program\_type 731

test\_id 725

is\_handicapped 718

gender 707

difficulty\_level 703

education 702

test\_type 702

city\_tier 700

program\_id 699

total\_programs\_enrolled 692

program\_duration 675

is\_pass 0

id\_num 0

dtype: int64

In [24]:

print(train\_data['difficulty\_level'].unique())

print(train\_data['education'].unique())

print(train\_data['total\_programs\_enrolled'].unique())

print(train\_data['city\_tier'].unique())

print(train\_data['is\_handicapped'].unique())

print(train\_data['age'].unique())

print(train\_data['trainee\_engagement\_rating'].unique())

print(train\_data['gender'].unique())

print(train\_data['program\_duration'].unique())

['intermediate' 'easy' 'hard' 'vary hard' nan]

['Matriculation' 'High School Diploma' 'Bachelors' nan 'Masters'

'No Qualification']

[ 5. 2. 1. 4. 3. nan 6. 7. 10. 14. 8. 9. 12. 11.]

[ 3. 4. 1. 2. nan]

['N' 'Y' nan]

[24. 26. 40. nan 42. 29. 48. 45. 28. 43. 27. 32. 44. 41. 23. 38. 30. 46.

25. 31. 34. 56. 47. 33. 22. 59. 49. 21. 18. 35. 36. 50. 55. 39. 51. 20.

19. 54. 58. 57. 37. 52. 53. 17. 63. 60.]

[ 1. 3. 2. 4. 5. nan]

['M' 'F' nan]

[136. 131. 120. 117. 134. 135. 121. 133. 122. 130. nan]

In [25]:

percent **=**(train\_data.isnull().sum()**/**train\_data.isnull().count()).sort\_values(ascending**=False**)

missing\_data **=** pd.concat([total, percent], axis**=**1, keys**=**['Total', 'Percent'])

missing\_data*#.head(20)*

Out[25]:

|  | **Total** | **Percent** |
| --- | --- | --- |
| **age** | 19379 | 0.387596 |
| **trainee\_engagement\_rating** | 772 | 0.015441 |
| **trainee\_id** | 739 | 0.014781 |
| **program\_type** | 731 | 0.014621 |
| **test\_id** | 725 | 0.014501 |
| **is\_handicapped** | 718 | 0.014361 |
| **gender** | 707 | 0.014141 |
| **difficulty\_level** | 703 | 0.014061 |
| **education** | 702 | 0.014041 |
| **test\_type** | 702 | 0.014041 |
| **city\_tier** | 700 | 0.014001 |
| **program\_id** | 699 | 0.013981 |
| **total\_programs\_enrolled** | 692 | 0.013841 |
| **program\_duration** | 675 | 0.013501 |
| **is\_pass** | 0 | 0.000000 |
| **id\_num** | 0 | 0.000000 |

# **数据预处理**

## **热编码**

In [26]:

*# loc定位到目标行，对gender特征进行独热编码*

train\_data.loc[train\_data['gender'] **==** 'F', 'gender'] **=** 0 *# 令gender等于F那行的gender值为0*

train\_data.loc[train\_data['gender'] **==** 'M', 'gender'] **=** 1 *# 令gender等于M那行的gender值为1*

​

train\_data.loc[train\_data['is\_handicapped'] **==** 'N', 'is\_handicapped'] **=** 0

train\_data.loc[train\_data['is\_handicapped'] **==** 'Y', 'is\_handicapped'] **=** 1

​

train\_data.loc[train\_data['education'] **==** 'No Qualification', 'education'] **=** 0*# 将education转换成值*

train\_data.loc[train\_data['education'] **==** 'High School Diploma', 'education'] **=** 2

train\_data.loc[train\_data['education'] **==** 'Matriculation', 'education'] **=** 4

train\_data.loc[train\_data['education'] **==** 'Bachelors', 'education'] **=** 6

train\_data.loc[train\_data['education'] **==** 'Masters', 'education'] **=** 8

​

train\_data.loc[train\_data['difficulty\_level'] **==** 'easy', 'difficulty\_level'] **=** 0 *#将level转换成值*

train\_data.loc[train\_data['difficulty\_level'] **==** 'intermediate', 'difficulty\_level'] **=** 2

train\_data.loc[train\_data['difficulty\_level'] **==** 'hard', 'difficulty\_level'] **=** 4

train\_data.loc[train\_data['difficulty\_level'] **==** 'vary hard', 'difficulty\_level'] **=** 6

​

train\_data.loc[train\_data['test\_type'] **==** 'offline', 'test\_type'] **=** 0 *#将test\_type转换成值*

train\_data.loc[train\_data['test\_type'] **==** 'online', 'test\_type'] **=** 1

​

​

## **填充及删除缺失行**

In [27]:

print(train\_data['age'].mean())

print(train\_data['trainee\_engagement\_rating'].mean())

print(train\_data['city\_tier'].mean())

print(train\_data['education'].mean())

print(train\_data['total\_programs\_enrolled'].mean())

print(train\_data['program\_duration'].mean())

print(train\_data['difficulty\_level'].mean())

36.51425585420817

2.3978182261406573

2.249097326463548

3.377555988315482

2.583113617004016

128.22936561036434

1.3008621564053149

In [28]:

train\_data['age'] **=** train\_data['age'].fillna(train\_data['age'].mean())

train\_data['gender'] **=** train\_data['gender'].fillna(train\_data['gender'].mean())

train\_data['trainee\_engagement\_rating'] **=** train\_data['trainee\_engagement\_rating'].fillna(train\_data['trainee\_engagement\_rating'].mean())

train\_data['city\_tier'] **=** train\_data['city\_tier'].fillna(train\_data['city\_tier'].mean())

train\_data['education'] **=** train\_data['education'].fillna(train\_data['education'].mean())

*#train['program\_type'] = train['program\_type'].fillna('Y')#Y最多*

train\_data['test\_type'] **=** train\_data['test\_type'].fillna(train\_data['test\_type'].mode()[0])*#线下较多*

train\_data['total\_programs\_enrolled'] **=** train\_data['total\_programs\_enrolled'].fillna(train\_data['total\_programs\_enrolled'].mean())

train\_data['program\_duration'] **=** train\_data['program\_duration'].fillna(train\_data['program\_duration'].mean())

train\_data['is\_handicapped'] **=** train\_data['is\_handicapped'].fillna(0)*#大部分人不残疾*

train\_data['difficulty\_level'] **=** train\_data['difficulty\_level'].fillna(train\_data['difficulty\_level'].mean())

train\_data.dropna(inplace**=True**,axis**=**0,subset **=** ["program\_type","trainee\_id","test\_id","program\_id"])

## **处理后查看数据**

In [29]:

total **=** train\_data.isnull().sum().sort\_values(ascending**=False**)

print(total)

is\_pass 0

trainee\_engagement\_rating 0

is\_handicapped 0

total\_programs\_enrolled 0

age 0

city\_tier 0

education 0

gender 0

trainee\_id 0

difficulty\_level 0

test\_type 0

test\_id 0

program\_duration 0

program\_id 0

program\_type 0

id\_num 0

dtype: int64

In [30]:

print(train\_data['difficulty\_level'].unique())

print(train\_data['education'].unique())

print(train\_data['total\_programs\_enrolled'].unique())

print(train\_data['city\_tier'].unique())

print(train\_data['is\_handicapped'].unique())

print(train\_data['age'].unique())

print(train\_data['trainee\_engagement\_rating'].unique())

print(train\_data['gender'].unique())

print(train\_data['program\_duration'].unique())

[2. 0. 4. 6. 1.30086216]

[4. 2. 6. 3.37755599 8. 0. ]

[ 5. 2. 1. 4. 3. 2.58311362

6. 7. 10. 8. 9. 12.

11. 14. ]

[3. 4. 1. 2. 2.24909733]

[0 1]

[24. 26. 40. 36.51425585 42. 29.

48. 28. 45. 43. 32. 44.

41. 23. 38. 27. 30. 46.

25. 31. 34. 56. 47. 33.

22. 59. 49. 21. 18. 35.

36. 50. 39. 51. 20. 19.

54. 58. 57. 37. 55. 52.

53. 17. 63. 60. ]

[1. 3. 2. 4. 5. 2.39781823]

[1. 0. 0.53307906]

[136. 131. 120. 117. 134.

135. 121. 133. 122. 130.

128.22936561]

# **特征工程**

## **线性回归**

In [31]:

**from** sklearn.linear\_model **import** LinearRegression *# 导入线性回归的类*

**from** sklearn.model\_selection **import** KFold *# 交叉验证*

​

*#predictors = ["trainee\_engagement\_rating","is\_handicapped", "total\_programs\_enrolled", "age", "gender", "difficulty\_level", "program\_duration", "city\_tier","education"] # 输入机器学习算法的特征*

predictors **=** ["trainee\_engagement\_rating","is\_handicapped", "total\_programs\_enrolled", "age", "gender", "difficulty\_level", "program\_duration", "city\_tier","education","test\_type"] *# 输入机器学习算法的特征*

alg **=** LinearRegression() *# 初始化线性回归模型*

alg1 = alg

kf **=** KFold(n\_splits**=**3,random\_state**=**2,shuffle**=True**) *# KFold类实例化*

​

predictions = []

MaxNum**=**0*#最大拟合数*

​

*# 有3次for循环，每次建立一个回归模型*

**for** train, test **in** kf.split(train\_data):

train\_predictors **=** (train\_data[predictors].iloc[train,:]) *# 取出训练数据，获得训练特征*

train\_target **=** train\_data["is\_pass"].iloc[train] *# 获取到数据集中交叉分类好的标签target，即是否通过 通过*

print(train\_predictors)

print(train\_target)

alg.fit(train\_predictors, train\_target) *# 通过特征和target获得训练模型*

print(train\_data[predictors].iloc[test,:])

test\_predictions **=** alg.predict(train\_data[predictors].iloc[test,:]) *# 判断模型准确度*

*# predictions = np.concatenate(predictions, axis=0) # 转换成数组，才能比较大小*

test\_predictions[test\_predictions**>**0.5]**=**1 *#转换test\_predictions的值*

test\_predictions[test\_predictions**<=**0.5]**=**0

print("预测结果：",test\_predictions)

print("真实结果：",test\_target)

print("正确的数量为：",sum(test\_predictions==test\_target))

print ("准确率为：",alg1.score(train\_data[predictors],train\_data['is\_pass']))

trainee\_engagement\_rating is\_handicapped total\_programs\_enrolled \

1 3.0 0 2.0

3 1.0 0 4.0

4 4.0 0 2.0

5 2.0 0 4.0

6 3.0 0 3.0

... ... ... ...

49989 1.0 1 4.0

49992 1.0 0 1.0

49993 4.0 0 5.0

49994 3.0 0 2.0

49996 5.0 0 2.0

age gender difficulty\_level program\_duration city\_tier \

1 26.000000 0.0 0.0 131.000000 4.0

3 36.514256 0.0 0.0 117.000000 3.0

4 42.000000 0.0 2.0 131.000000 1.0

5 29.000000 0.0 2.0 134.000000 1.0

6 48.000000 1.0 0.0 120.000000 2.0

... ... ... ... ... ...

49989 24.000000 0.0 0.0 120.000000 4.0

49992 26.000000 0.0 0.0 120.000000 4.0

49993 36.514256 0.0 4.0 121.000000 2.0

49994 30.000000 0.0 2.0 128.229366 3.0

49996 36.514256 1.0 2.0 121.000000 3.0

education test\_type

1 2.0 0

3 4.0 0

4 2.0 0

5 2.0 0

6 6.0 1

... ... ...

49989 4.0 1

49992 4.0 1

49993 2.0 0

49994 2.0 0

49996 4.0 0

[31449 rows x 10 columns]

1 1

3 1

4 1

5 0

6 1

..

49989 1

49992 1

49993 1

49994 1

49996 1

Name: is\_pass, Length: 31449, dtype: int64

Out[23]:

LinearRegression()

trainee\_engagement\_rating is\_handicapped total\_programs\_enrolled \

0 1.000000 0 5.0

2 2.000000 0 1.0

9 4.000000 0 2.0

12 4.000000 0 4.0

14 1.000000 0 2.0

... ... ... ...

49987 4.000000 1 4.0

49990 2.000000 0 1.0

49991 1.000000 0 3.0

49995 1.000000 0 2.0

49997 2.397818 0 5.0

age gender difficulty\_level program\_duration city\_tier \

0 24.000000 1.0 2.0 136.0 3.0

2 40.000000 1.0 0.0 120.0 1.0

9 28.000000 1.0 4.0 136.0 3.0

12 36.514256 1.0 2.0 136.0 1.0

14 36.514256 1.0 0.0 134.0 2.0

... ... ... ... ... ...

49987 36.514256 1.0 0.0 134.0 2.0

49990 45.000000 0.0 0.0 136.0 2.0

49991 26.000000 1.0 0.0 134.0 4.0

49995 41.000000 0.0 0.0 117.0 2.0

49997 29.000000 1.0 4.0 134.0 2.0

education test\_type

0 4.0 0

2 4.0 1

9 6.0 0

12 2.0 0

14 6.0 1

... ... ...

49987 2.0 1

49990 2.0 0

49991 6.0 1

49995 2.0 1

49997 4.0 0

[15725 rows x 10 columns]

预测结果： [0. 1. 1. ... 1. 1. 1.]

真实结果： 0 0

2 1

9 1

12 1

14 1

..

49987 1

49990 1

49991 0

49995 1

49997 1

Name: is\_pass, Length: 15725, dtype: int64

正确的数量为： 11017

准确率为： 0.233539661678043

trainee\_engagement\_rating is\_handicapped total\_programs\_enrolled \

0 1.000000 0 5.0

2 2.000000 0 1.0

3 1.000000 0 4.0

8 3.000000 0 3.0

9 4.000000 0 2.0

... ... ... ...

49991 1.000000 0 3.0

49992 1.000000 0 1.0

49993 4.000000 0 5.0

49995 1.000000 0 2.0

49997 2.397818 0 5.0

age gender difficulty\_level program\_duration city\_tier \

0 24.000000 1.0 2.0 136.0 3.0

2 40.000000 1.0 0.0 120.0 1.0

3 36.514256 0.0 0.0 117.0 3.0

8 36.514256 1.0 0.0 134.0 4.0

9 28.000000 1.0 4.0 136.0 3.0

... ... ... ... ... ...

49991 26.000000 1.0 0.0 134.0 4.0

49992 26.000000 0.0 0.0 120.0 4.0

49993 36.514256 0.0 4.0 121.0 2.0

49995 41.000000 0.0 0.0 117.0 2.0

49997 29.000000 1.0 4.0 134.0 2.0

education test\_type

0 4.0 0

2 4.0 1

3 4.0 0

8 4.0 0

9 6.0 0

... ... ...

49991 6.0 1

49992 4.0 1

49993 2.0 0

49995 2.0 1

49997 4.0 0

[31449 rows x 10 columns]

0 0

2 1

3 1

8 1

9 1

..

49991 0

49992 1

49993 1

49995 1

49997 1

Name: is\_pass, Length: 31449, dtype: int64

Out[23]:

LinearRegression()

trainee\_engagement\_rating is\_handicapped total\_programs\_enrolled \

1 3.0 0 2.0

4 4.0 0 2.0

5 2.0 0 4.0

6 3.0 0 3.0

11 1.0 0 2.0

... ... ... ...

49983 3.0 0 2.0

49986 1.0 0 2.0

49988 4.0 0 2.0

49994 3.0 0 2.0

49996 5.0 0 2.0

age gender difficulty\_level program\_duration city\_tier \

1 26.000000 0.0 0.0 131.000000 4.0

4 42.000000 0.0 2.0 131.000000 1.0

5 29.000000 0.0 2.0 134.000000 1.0

6 48.000000 1.0 0.0 120.000000 2.0

11 36.514256 0.0 0.0 134.000000 4.0

... ... ... ... ... ...

49983 46.000000 0.0 2.0 134.000000 1.0

49986 45.000000 0.0 0.0 134.000000 1.0

49988 27.000000 1.0 0.0 136.000000 2.0

49994 30.000000 0.0 2.0 128.229366 3.0

49996 36.514256 1.0 2.0 121.000000 3.0

education test\_type

1 2.0 0

4 2.0 0

5 2.0 0

6 6.0 1

11 2.0 1

... ... ...

49983 2.0 0

49986 4.0 1

49988 4.0 1

49994 2.0 0

49996 4.0 0

[15725 rows x 10 columns]

预测结果： [1. 1. 1. ... 1. 1. 1.]

真实结果： 1 1

4 1

5 0

6 1

11 1

..

49983 0

49986 1

49988 1

49994 1

49996 1

Name: is\_pass, Length: 15725, dtype: int64

正确的数量为： 11047

准确率为： 0.2341756052062577

trainee\_engagement\_rating is\_handicapped total\_programs\_enrolled \

0 1.000000 0 5.0

1 3.000000 0 2.0

2 2.000000 0 1.0

4 4.000000 0 2.0

5 2.000000 0 4.0

... ... ... ...

49991 1.000000 0 3.0

49994 3.000000 0 2.0

49995 1.000000 0 2.0

49996 5.000000 0 2.0

49997 2.397818 0 5.0

age gender difficulty\_level program\_duration city\_tier \

0 24.000000 1.0 2.0 136.000000 3.0

1 26.000000 0.0 0.0 131.000000 4.0

2 40.000000 1.0 0.0 120.000000 1.0

4 42.000000 0.0 2.0 131.000000 1.0

5 29.000000 0.0 2.0 134.000000 1.0

... ... ... ... ... ...

49991 26.000000 1.0 0.0 134.000000 4.0

49994 30.000000 0.0 2.0 128.229366 3.0

49995 41.000000 0.0 0.0 117.000000 2.0

49996 36.514256 1.0 2.0 121.000000 3.0

49997 29.000000 1.0 4.0 134.000000 2.0

education test\_type

0 4.0 0

1 2.0 0

2 4.0 1

4 2.0 0

5 2.0 0

... ... ...

49991 6.0 1

49994 2.0 0

49995 2.0 1

49996 4.0 0

49997 4.0 0

[31450 rows x 10 columns]

0 0

1 1

2 1

4 1

5 0

..

49991 0

49994 1

49995 1

49996 1

49997 1

Name: is\_pass, Length: 31450, dtype: int64

Out[23]:

LinearRegression()

trainee\_engagement\_rating is\_handicapped total\_programs\_enrolled \

3 1.0 0 4.000000

8 3.0 0 3.000000

10 4.0 0 2.000000

13 5.0 0 4.000000

15 2.0 1 2.583114

... ... ... ...

49980 1.0 1 2.000000

49982 1.0 0 4.000000

49989 1.0 1 4.000000

49992 1.0 0 1.000000

49993 4.0 0 5.000000

age gender difficulty\_level program\_duration city\_tier \

3 36.514256 0.0 0.0 117.0 3.0

8 36.514256 1.0 0.0 134.0 4.0

10 36.514256 0.0 0.0 131.0 3.0

13 36.514256 1.0 2.0 135.0 3.0

15 45.000000 1.0 6.0 134.0 2.0

... ... ... ... ... ...

49980 26.000000 1.0 0.0 130.0 2.0

49982 24.000000 1.0 2.0 134.0 3.0

49989 24.000000 0.0 0.0 120.0 4.0

49992 26.000000 0.0 0.0 120.0 4.0

49993 36.514256 0.0 4.0 121.0 2.0

education test\_type

3 4.0 0

8 4.0 0

10 2.0 0

13 4.0 0

15 2.0 0

... ... ...

49980 4.0 0

49982 2.0 0

49989 4.0 1

49992 4.0 1

49993 2.0 0

[15724 rows x 10 columns]

预测结果： [1. 1. 1. ... 1. 1. 1.]

真实结果： 3 1

8 1

10 1

13 1

15 0

..

49980 1

49982 1

49989 1

49992 1

49993 1

Name: is\_pass, Length: 15724, dtype: int64

正确的数量为： 11150

准确率为： 0.06562675479072799

## **逻辑回归**

In [63]:

**from** sklearn **import** model\_selection

**from** sklearn.linear\_model **import** LogisticRegression

​

alg **=** LogisticRegression(random\_state**=**1, solver**=**'liblinear') *# 初始化逻辑回归类*

alg2 **=** alg

alg2.fit(train\_predictors, train\_target) *# 通过特征和target获得训练模型*

​

*# 逻辑回归交叉验证*

score **=** model\_selection.cross\_val\_score(alg2, train\_data[predictors], train\_data["is\_pass"], cv**=**3)

print("准确率为：", score.mean())

Out[63]:

LogisticRegression(random\_state=1, solver='liblinear')

准确率为： 0.7069572003655532

## **测试数据导入**

In [32]:

test1 **=** pd.read\_csv("d:/工作室/QG/train and test2/test2.csv")

test1.head(10)

total **=** test1.isnull().sum().sort\_values(ascending**=False**)

print(total)

​

Out[32]:

|  | **id\_num** | **program\_type** | **program\_id** | **program\_duration** | **test\_id** | **test\_type** | **difficulty\_level** | **trainee\_id** | **gender** | **education** | **city\_tier** | **age** | **total\_programs\_enrolled** | **is\_handicapped** | **trainee\_engagement\_rating** | **is\_pass** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 4753\_31 | T | T\_3 | 134 | 31 | online | easy | 4753 | F | Bachelors | 1 | 31.0 | 4 | N | 2.0 | NaN |
| **1** | 10816\_148 | Y | Y\_2 | 120 | 148 | online | easy | 10816 | M | Matriculation | 3 | NaN | 2 | N | 4.0 | NaN |
| **2** | 7498\_114 | Y | Y\_4 | 121 | 114 | offline | intermediate | 7498 | M | High School Diploma | 1 | NaN | 3 | N | 1.0 | NaN |
| **3** | 14139\_175 | Z | Z\_2 | 120 | 175 | online | easy | 14139 | F | High School Diploma | 1 | 44.0 | 1 | Y | 1.0 | NaN |
| **4** | 11305\_25 | T | T\_3 | 134 | 25 | offline | intermediate | 11305 | M | High School Diploma | 2 | 49.0 | 2 | N | 2.0 | NaN |
| **5** | 19047\_14 | T | T\_4 | 120 | 14 | offline | intermediate | 19047 | F | Bachelors | 4 | 44.0 | 4 | N | 1.0 | NaN |
| **6** | 19335\_164 | Z | Z\_3 | 130 | 164 | online | easy | 19335 | F | Matriculation | 4 | 26.0 | 1 | N | 2.0 | NaN |
| **7** | 13935\_38 | T | T\_2 | 117 | 38 | online | easy | 13935 | F | Matriculation | 3 | NaN | 2 | N | 1.0 | NaN |
| **8** | 6412\_29 | T | T\_3 | 134 | 29 | online | easy | 6412 | F | High School Diploma | 4 | 28.0 | 2 | N | 1.0 | NaN |
| **9** | 5615\_20 | T | T\_4 | 120 | 20 | online | easy | 5615 | F | Matriculation | 4 | 30.0 | 4 | N | 3.0 | NaN |

is\_pass 11465

age 4397

trainee\_engagement\_rating 14

is\_handicapped 0

total\_programs\_enrolled 0

city\_tier 0

education 0

gender 0

trainee\_id 0

difficulty\_level 0

test\_type 0

test\_id 0

program\_duration 0

program\_id 0

program\_type 0

id\_num 0

dtype: int64

## **测试数据预处理**

In [33]:

test1.loc[test1['gender'] **==** 'F', 'gender'] **=** 0 *# 令gender等于F那行的gender值为0*

test1.loc[test1['gender'] **==** 'M', 'gender'] **=** 1 *# 令gender等于M那行的gender值为1*

​

test1.loc[test1['is\_handicapped'] **==** 'N', 'is\_handicapped'] **=** 0

test1.loc[test1['is\_handicapped'] **==** 'Y', 'is\_handicapped'] **=** 1

​

test1.loc[test1['education'] **==** 'No Qualification', 'education'] **=** 0*# 将education转换成值*

test1.loc[test1['education'] **==** 'High School Diploma', 'education'] **=** 2

test1.loc[test1['education'] **==** 'Matriculation', 'education'] **=** 4

test1.loc[test1['education'] **==** 'Bachelors', 'education'] **=** 6

test1.loc[test1['education'] **==** 'Masters', 'education'] **=** 8

​

test1.loc[test1['difficulty\_level'] **==** 'easy', 'difficulty\_level'] **=** 0 *#将level转换成值*

test1.loc[test1['difficulty\_level'] **==** 'intermediate', 'difficulty\_level'] **=** 2

test1.loc[test1['difficulty\_level'] **==** 'hard', 'difficulty\_level'] **=** 4

test1.loc[test1['difficulty\_level'] **==** 'vary hard', 'difficulty\_level'] **=** 6

​

​

test1.loc[test1['test\_type'] **==** 'offline', 'test\_type'] **=** 0

test1.loc[test1['test\_type'] **==** 'online', 'test\_type'] **=** 1

In [34]:

test1['age'] **=** test1['age'].fillna(test1['age'].mean())

*#test1['gender'] = test1['gender'].fillna(test1['gender'].mean())*

test1['trainee\_engagement\_rating'] **=** test1['trainee\_engagement\_rating'].fillna(test1['trainee\_engagement\_rating'].mean())

*#test1['city\_tier'] = test1['city\_tier'].fillna(test1['city\_tier'].median())*

*#test1['education'] = test1['education'].fillna(test1['education'].mean())*

*#train['program\_type'] = train['program\_type'].fillna('Y')#Y最多*

*#test1['test\_type'] = test1['test\_type'].fillna('offline')#线下较多*

*#test1['total\_programs\_enrolled'] = test1['total\_programs\_enrolled'].fillna(test1['total\_programs\_enrolled'].mean())*

*#test1['program\_duration'] = test1['program\_duration'].fillna(test1['program\_duration'].mean())*

*#test1['is\_handicapped'] = test1['is\_handicapped'].fillna(0)#大部分人不残疾*

*#test1['difficulty\_level'] = test1['difficulty\_level'].fillna(test1['difficulty\_level'].mean())*

*#test1.dropna(inplace=True,axis=0,subset = ["program\_type","trainee\_id","test\_id","program\_id"])*

In [35]:

test1\_target**=**alg1.predict(test1[predictors])

print(test1\_target)

test1\_target[test1\_target**>**0.5]**=**1

test1\_target[test1\_target**<=**0.5]**=**0

print(test1\_target)

[0.75497728 0.86407892 0.5783156 ... 0.81746907 0.80153897 0.66642759]

[1. 1. 1. ... 1. 1. 1.]

## **测试数据导出**

In [36]:

out**=**pd.read\_csv("d:/工作室/QG/train and test2/submission2.csv")*#预览submission2*

print (sub)

out['is\_pass']**=**test1\_target*#将is\_pass数据赋值给submission2*

sub.rename\_axis(index**=**'')

print(out)

out.to\_csv("d:/工作室/QG/train and test2/submission3.csv",index**=None**)*#输出为文件submission3.csv*

id\_num is\_pass

0 4753\_31 0

1 10816\_148 0

2 7498\_114 0

3 14139\_175 0

4 11305\_25 0

... ... ...

11460 1612\_108 0

11461 17726\_44 0

11462 8272\_131 0

11463 10621\_166 0

11464 2465\_48 0

[11465 rows x 2 columns]

Out[36]:

|  | **id\_num** | **is\_pass** |
| --- | --- | --- |
|  |  |  |
| **0** | 4753\_31 | 1.0 |
| **1** | 10816\_148 | 1.0 |
| **2** | 7498\_114 | 1.0 |
| **3** | 14139\_175 | 1.0 |
| **4** | 11305\_25 | 1.0 |
| **...** | ... | ... |
| **11460** | 1612\_108 | 1.0 |
| **11461** | 17726\_44 | 1.0 |
| **11462** | 8272\_131 | 1.0 |
| **11463** | 10621\_166 | 1.0 |
| **11464** | 2465\_48 | 1.0 |

11465 rows × 2 columns

id\_num is\_pass

0 4753\_31 1.0

1 10816\_148 1.0

2 7498\_114 1.0

3 14139\_175 1.0

4 11305\_25 1.0

... ... ...

11460 1612\_108 1.0

11461 17726\_44 1.0

11462 8272\_131 1.0

11463 10621\_166 1.0

11464 2465\_48 1.0

[11465 rows x 2 columns]