In [1]:

```
%matplotlib inline
import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
plt.rc("font", size=14)
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
import seaborn as sns

df = pd.read_csv("./data/bank-additional-full.csv", header=0,nrows =3999)
df = df.dropna()
print(df.shape)
print(list(df.columns))

df.head()
```

```
(3999, 21)
['age', 'job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'month', 'day_of_week', 'duration', 'campaign', 'pday
s', 'previous', 'poutcome', 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed', 'y']
```

Out[1]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcome	emp.var.rate
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	...	1	999	0	nonexistent	1.1
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	...	1	999	0	nonexistent	1.1
2	37	services	married	high.school	no	yes	no	telephone	may	mon	...	1	999	0	nonexistent	1.1
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	...	1	999	0	nonexistent	1.1
4	56	services	married	high.school	no	no	yes	telephone	may	mon	...	1	999	0	nonexistent	1.1

5 rows × 21 columns

In [2]:

```
df=df.sample(n=3999)
```

In [3]:

```
df
```

Out[3]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcome
3701	51	admin.	single	basic.6y	no	yes	no	telephone	may	fri	...	1	999	0	nonexistent
3479	46	blue-collar	married	basic.6y	unknown	yes	no	telephone	may	thu	...	9	999	0	nonexistent
719	41	blue-collar	married	basic.4y	no	yes	no	telephone	may	tue	...	2	999	0	nonexistent
3300	36	blue-collar	married	basic.4y	no	no	no	telephone	may	thu	...	2	999	0	nonexistent
485	36	admin.	married	university.degree	no	unknown	unknown	telephone	may	tue	...	2	999	0	nonexistent
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1858	55	unemployed	single	basic.4y	unknown	unknown	unknown	telephone	may	fri	...	7	999	0	nonexistent
3739	32	blue-collar	married	basic.9y	no	no	no	telephone	may	fri	...	2	999	0	nonexistent
2618	33	admin.	single	university.degree	no	yes	no	telephone	may	tue	...	2	999	0	nonexistent
2322	48	admin.	divorced	high.school	no	no	yes	telephone	may	tue	...	2	999	0	nonexistent
1691	33	blue-collar	married	basic.6y	unknown	yes	yes	telephone	may	fri	...	2	999	0	nonexistent

3999 rows × 21 columns

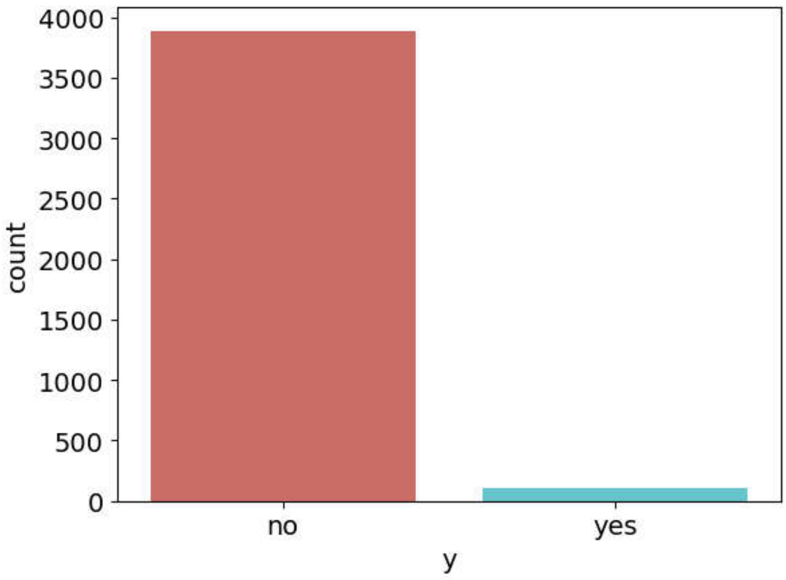
In [4]:

```
print(df['y'].value_counts())
print(df['y'].value_counts()/len(df))
```

```
no     3888
yes     111
Name: y, dtype: int64
no     0.972243
yes    0.027757
Name: y, dtype: float64
```

In [5]:

```
sns.countplot(x='y',data=df, palette='hls')
plt.show()
```



In [6]:

```
df.groupby('y').mean()
```

Out[6]:

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
y										
no	40.761317	249.978138	2.248457	999.0	0.0	1.1	93.994	-36.4	4.85713	5191.0
yes	40.027027	989.207207	2.108108	999.0	0.0	1.1	93.994	-36.4	4.85745	5191.0

In [7]:

```
df
```

Out[7]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcome
3701	51	admin.	single	basic.6y	no	yes	no	telephone	may	fri	...	1	999	0	nonexistent
3479	46	blue-collar	married	basic.6y	unknown	yes	no	telephone	may	thu	...	9	999	0	nonexistent
719	41	blue-collar	married	basic.4y	no	yes	no	telephone	may	tue	...	2	999	0	nonexistent
3300	36	blue-collar	married	basic.4y	no	no	no	telephone	may	thu	...	2	999	0	nonexistent
485	36	admin.	married	university.degree	no	unknown	unknown	telephone	may	tue	...	2	999	0	nonexistent
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1858	55	unemployed	single	basic.4y	unknown	unknown	unknown	telephone	may	fri	...	7	999	0	nonexistent
3739	32	blue-collar	married	basic.9y	no	no	no	telephone	may	fri	...	2	999	0	nonexistent
2618	33	admin.	single	university.degree	no	yes	no	telephone	may	tue	...	2	999	0	nonexistent
2322	48	admin.	divorced	high.school	no	no	yes	telephone	may	tue	...	2	999	0	nonexistent
1691	33	blue-collar	married	basic.6y	unknown	yes	yes	telephone	may	fri	...	2	999	0	nonexistent

3999 rows × 21 columns

In [8]:

```
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
label = LabelEncoder()
for dataset in [df]:
    dataset['job_Code'] = label.fit_transform(dataset['job'])
    dataset['marital_Code'] = label.fit_transform(dataset['marital'])
    dataset['education_Code'] = label.fit_transform(dataset['education'])
    dataset['default_Code'] = label.fit_transform(dataset['default'])
    dataset['housing_Code'] = label.fit_transform(dataset['housing'])
    dataset['loan_Code'] = label.fit_transform(dataset['loan'])
    dataset['contact_Code'] = label.fit_transform(dataset['contact'])
    dataset['month_Code'] = label.fit_transform(dataset['month'])
    dataset['day_of_week_Code'] = label.fit_transform(dataset['day_of_week'])
```

In [9]:

```
dataset
```

Out[9]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	y	job_Code	marital_Code	educatior
3701	51	admin.	single	basic.6y	no	yes	no	telephone	may	fri	...	no	0	2	
3479	46	blue-collar	married	basic.6y	unknown	yes	no	telephone	may	thu	...	no	1	1	
719	41	blue-collar	married	basic.4y	no	yes	no	telephone	may	tue	...	no	1	1	
3300	36	blue-collar	married	basic.4y	no	no	no	telephone	may	thu	...	no	1	1	
485	36	admin.	married	university.degree	no	unknown	unknown	telephone	may	tue	...	no	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1858	55	unemployed	single	basic.4y	unknown	unknown	unknown	telephone	may	fri	...	no	10	2	
3739	32	blue-collar	married	basic.9y	no	no	no	telephone	may	fri	...	no	1	1	
2618	33	admin.	single	university.degree	no	yes	no	telephone	may	tue	...	no	0	2	
2322	48	admin.	divorced	high.school	no	no	yes	telephone	may	tue	...	no	0	0	
1691	33	blue-collar	married	basic.6y	unknown	yes	yes	telephone	may	fri	...	yes	1	1	

3999 rows × 30 columns

In [10]:

```
columns_train_data_x = [
    'age', 'job_Code', 'marital_Code', 'education_Code',
    'default_Code', 'housing_Code', 'loan_Code',
    'contact_Code', 'month_Code', 'day_of_week_Code',
    'duration','campaign','pdays','previous','poutcome','emp.var.rate','cons.price.idx','cons.conf.idx','euribor3m','nr.employed'

]
train_data_x = dataset[columns_train_data_x]
train_data_y = dataset['y']
# train_data_x = pd.get_dummies(train_data_x , columns=columns_train_data_x)
```

In [11]:

```
train_data_y
```

Out[11]:

```
3701     no
3479     no
719      no
3300     no
485      no
        ...
1858     no
3739     no
2618     no
2322     no
1691    yes
Name: y, Length: 3999, dtype: object
```

In [12]:

```
train_data_y=train_data_y.map(dict(yes=1, no=0))
```

In [13]:

```
train_data_x
```

Out[13]:

	age	job_Code	marital_Code	education_Code	default_Code	housing_Code	loan_Code	contact_Code	month_Code	day_of_week_Code	duration
3701	51	0	2	1	0	2	0	0	0	0	161
3479	46	1	1	1	1	2	0	0	0	2	516
719	41	1	1	0	0	2	0	0	0	3	1529
3300	36	1	1	0	0	0	0	0	0	2	544
485	36	0	1	5	0	1	1	0	0	3	176
...	...	...	...	...	...	...	...	...	...	...	...
1858	55	10	2	0	1	1	1	0	0	0	147
3739	32	1	1	2	0	0	0	0	0	0	200
2618	33	0	2	5	0	2	0	0	0	3	76
2322	48	0	0	3	0	0	2	0	0	3	284
1691	33	1	1	1	1	2	2	0	0	0	1132

3999 rows × 20 columns

In [14]:

```
train_data_x=train_data_x.fillna(0)
train_data_x=train_data_x.replace({"poutcome":{'nonexistent':0}})
```

In [ ]:

In [15]:

```
from sklearn.preprocessing import StandardScaler

ss = StandardScaler() ##
#用测试集训练并标准化
ss.fit(train_data_x)
train_data_x = ss.transform(train_data_x)
```

In [16]:

```
train_data_x
```

Out[16]:

```
array([[ 1.15993516e+00, -1.03202635e+00,  1.59799302e+00, ...,
        -1.42108547e-14,  1.10301918e+00,  0.00000000e+00],
       [ 5.94613083e-01, -7.45920745e-01, -1.73174964e-01, ...,
        -1.42108547e-14,  1.69557341e+00,  0.00000000e+00],
       [ 2.92910063e-02, -7.45920745e-01, -1.73174964e-01, ...,
        -1.42108547e-14, -8.20892835e-02,  0.00000000e+00],
       ...,
       [-8.75224316e-01, -1.03202635e+00,  1.59799302e+00, ...,
        -1.42108547e-14, -6.74643516e-01,  0.00000000e+00],
       [ 8.20741913e-01, -1.03202635e+00, -1.94434294e+00, ...,
        -1.42108547e-14, -6.74643516e-01,  0.00000000e+00],
       [-8.75224316e-01, -7.45920745e-01, -1.73174964e-01, ...,
        -1.42108547e-14, -1.26719775e+00,  0.00000000e+00]])
```

In [ ]:

In [17]:

```
X_train, X_test, y_train, y_test = train_test_split(train_data_x,train_data_y, test_size=0.3, random_state=0)
```

In [18]:

```
from xgboost.sklearn import XGBClassifier as xgb
```

In [19]:

```
import xgboost as xgb
```

In [27]:

```
#XGBoost训练预测得分
xg_classifier = xgb.XGBClassifier(objective ='binary:logistic', colsample_bytree = 0.3, learning_rate = 0.2,
                max_depth = 6, alpha = 10, n_estimators = 10)

xg_classifier.fit(X_train, y_train)
xg_classifier.score(X_test, y_test)
```

```
[17:19:14] WARNING: C:\Windows\Temp\abs_557yfx631l\croots\recipe\xgboost-split_1659548953302\work\src\learner.cc:1115: Starting in XGBo
ost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly se
t eval_metric if you'd like to restore the old behavior.

D:\Program\anaconda3\envs\graduate\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in XGBClassifier is
deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False wh
en constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
```

Out[27]:

```
0.9641666666666666
```

In [28]:

```
print('在测试数据集上面的预测准确率: {:.2f}'.format(xg_classifier.score(X_test, y_test)))
print ("\n\n ---XGB---")
rf_roc_auc = roc_auc_score(y_test, xg_classifier.predict(X_test))
print ("逻辑回归 AUC = %2.2f" % rf_roc_auc)
print(classification_report(y_test, xg_classifier.predict(X_test)))


#绘制Roc曲线观察模型的性能
fprl_gnb, tprl_gnb, thresholdsl_gnb = roc_curve(y_test, xg_classifier.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fprl_gnb, tprl_gnb, color = 'yellow',label='XGB Model  (area = %0.2f)' % rf_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('XGB_ROC')
plt.show()
```

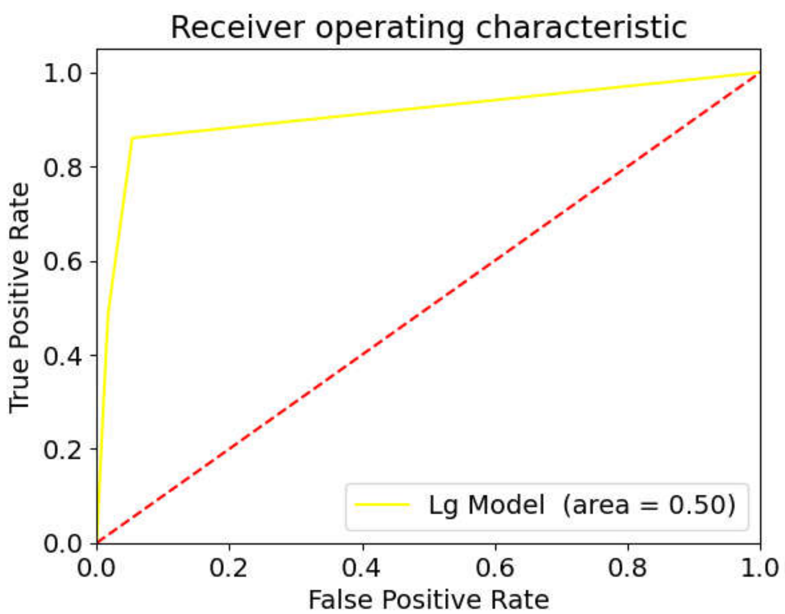
```
在测试数据集上面的预测准确率: 0.96


 ---XGB---
逻辑回归 AUC = 0.50
              precision    recall  f1-score   support

           0       0.96      1.00      0.98      1157
           1       0.00      0.00      0.00        43

    accuracy                           0.96      1200
   macro avg       0.48      0.50      0.49      1200
weighted avg       0.93      0.96      0.95      1200


D:\Program\anaconda3\envs\graduate\lib\site-packages\sklearn\metrics\_classification.py:1334: UndefinedMetricWarning: Precision and F-s
core are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
D:\Program\anaconda3\envs\graduate\lib\site-packages\sklearn\metrics\_classification.py:1334: UndefinedMetricWarning: Precision and F-s
core are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
D:\Program\anaconda3\envs\graduate\lib\site-packages\sklearn\metrics\_classification.py:1334: UndefinedMetricWarning: Precision and F-s
core are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```



In [29]:

```
#在测试集上进行预测
y_pred = xg_classifier.predict(X_test)
print('在测试集上预测的准确率: {:.2f}'.format(xg_classifier.score(X_test, y_test)))
```

```
在测试集上预测的准确率: 0.96
```

In [30]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
```

```
[[1157    0]
 [  43    0]]
```

In [32]:

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98      1157
           1       0.00      0.00      0.00        43

    accuracy                           0.96      1200
   macro avg       0.48      0.50      0.49      1200
weighted avg       0.93      0.96      0.95      1200

D:\Program\anaconda3\envs\graduate\lib\site-packages\sklearn\metrics\_classification.py:1334: UndefinedMetricWarning: Precision and F-s
core are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
D:\Program\anaconda3\envs\graduate\lib\site-packages\sklearn\metrics\_classification.py:1334: UndefinedMetricWarning: Precision and F-s
core are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
D:\Program\anaconda3\envs\graduate\lib\site-packages\sklearn\metrics\_classification.py:1334: UndefinedMetricWarning: Precision and F-s
core are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]: