In [ ]:

```
#功能分工：
#特征工程：谢栓虎、李博
#模型解释：张彬彬
#模型结果：张官喜
```

In [5]:

```
#特征工程
import numpy as np
import pandas as pd
```

In [5]:

```
# 加载数据
# train_01 = pd.read_csv('E:/02Work/AI/12DAY09-180512-自由讨论/数据集/train01.csv')
train_02 =pd.read_csv('E:/AI/光环AI/课件/项目/优惠券/data2/train02.csv', header=0)
```

In [3]:

```
# train_01.shape
```

In [6]:

```
# 查看数据
train_02.head(10)
```

Out[6]:

	user_id	discount_rate	distance	day_of_month	days_distance	discount_man	disc
0	1439408	0.866667	1.0	28	14	150.0	20.0
1	1439408	0.866667	1.0	28	14	150.0	20.0
2	1439408	0.950000	0.0	13	30	20.0	1.0
3	1439408	0.950000	0.0	13	30	20.0	1.0
4	1439408	0.950000	0.0	16	2	20.0	1.0
5	1439408	0.950000	0.0	16	2	20.0	1.0
6	2029232	0.833333	0.0	30	16	30.0	5.0
7	2029232	0.833333	0.0	30	16	30.0	5.0
8	2029232	0.950000	0.0	19	5	20.0	1.0
9	2747744	0.800000	NaN	6	23	50.0	10.0

10 rows × 56 columns

In [ ]:

```
train_02.shape
```

In [6]:

```
train_02.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 255225 entries, 0 to 255224
Data columns (total 56 columns):
user_id                                  255225 non-null int64
discount_rate                            255225 non-null float64
distance                                 228623 non-null float64
day_of_month                             255225 non-null int64
days_distance                            255225 non-null int64
discount_man                             247356 non-null float64
discount_jian                            247356 non-null float64
is_man_jian                              255225 non-null int64
total_sales                              250077 non-null float64
sales_use_coupon                         251599 non-null float64
total_coupon                             251599 non-null float64
merchant_min_distance                    193677 non-null float64
merchant_max_distance                    193677 non-null float64
merchant_mean_distance                   193677 non-null float64
merchant_median_distance                 193677 non-null float64
merchant_coupon_transfer_rate            216615 non-null float64
coupon_rate                              250077 non-null float64
count_merchant                           134766 non-null float64
user_min_distance                        14974 non-null float64
user_max_distance                        14974 non-null float64
user_mean_distance                       14974 non-null float64
user_median_distance                     14974 non-null float64
buy_use_coupon                           134766 non-null float64
buy_total                                134766 non-null float64
coupon_received                          134766 non-null float64
avg_user_date_datereceived_gap           16187 non-null float64
min_user_date_datereceived_gap           16187 non-null float64
max_user_date_datereceived_gap           16187 non-null float64
buy_use_coupon_rate                      99643 non-null float64
user_coupon_transfer_rate                98346 non-null float64
user_merchant_buy_total                  255225 non-null float64
user_merchant_received                   255225 non-null float64
user_merchant_buy_use_coupon             75584 non-null float64
user_merchant_any                        255225 non-null float64
user_merchant_buy_common                 75584 non-null float64
user_merchant_coupon_transfer_rate       30751 non-null float64
user_merchant_coupon_buy_rate            64842 non-null float64
user_merchant_rate                       64842 non-null float64
user_merchant_common_buy_rate            64842 non-null float64
this_month_user_receive_same_coupon_count 255225 non-null int64
this_month_user_receive_all_coupon_count 255225 non-null int64
this_month_user_receive_same_coupon_lastone 255225 non-null int64
this_month_user_receive_same_coupon_firstone 255225 non-null int64
this_day_user_receive_all_coupon_count   255225 non-null int64
this_day_user_receive_same_coupon_count  255225 non-null int64
day_gap_before                           255225 non-null int64
day_gap_after                            255225 non-null int64
is_weekend                               255225 non-null int64
weekday1                                 255225 non-null int64
weekday2                                 255225 non-null int64
weekday3                                 255225 non-null int64
weekday4                                 255225 non-null int64
weekday5                                 255225 non-null int64
```

```
weekday6                                255225 non-null int64
weekday7                                255225 non-null int64
label                                   255225 non-null int64
dtypes: float64(35), int64(21)
memory usage: 109.0 MB
```

In [10]:

```
# train_02.describe
# 去除user_id
train_02.drop(['user_id'],axis=1,inplace=True)
# 去除标签中 -1的数据，会给后边处理造成问题，并且-1数量为3336，占比例较小
train_02.label[train_02.label == -1].value_counts()
train_02 = train_02[train_02.label > -1]
train_02.shape
```

Out[10]:

```
(486936, 55)
```

In [10]:

```
from sklearn import preprocessing
from sklearn import feature_selection
```

In [11]:

```
# 缺失值计算(也可用pandas.fillna函数)
imp = preprocessing.Imputer(missing_values='NaN', strategy='mean', axis=0)
train_02_new = imp.fit_transform(train_02)
# train_02 = pd.DataFrame(train_02_new)
```

In [12]:

```
train_02_data = train_02_new[:,:-1]
train_02_label = train_02_new[:,-1:]
```

In [13]:

```
# 特征选择，使用GBDT
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import GradientBoostingClassifier
```

In [14]:

```
selector = SelectFromModel(GradientBoostingClassifier()).fit(train_02_data, train_02_label)
data = selector.transform(train_02_data)
print(data)
print(selector.estimator_.feature_importances_)
```

```
C:\Program Files\Anaconda3\lib\site-packages\sklearn\utils\validation.py:578: DataCo
nversionWarning: A column-vector y was passed when a 1d array was expected. Please c
hange the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)

[[   0.9          16.          200.         ...,   -1.          -1.          -1.
      ]
 [   0.9          16.          200.         ...,   -1.          -1.          -1.
      ]
 [   0.85          8.          200.         ...,   -1.          -1.          -1.
      ]
 ...,
 [   0.96666667    2.           30.         ...,   -1.          -1.          -1.
      ]
 [   0.96666667    2.           30.         ...,   -1.          -1.          -1.
      ]
 [   0.96666667   21.           30.         ...,    1.          -1.           4.
      ]]
[ 0.05000658  0.01465191  0.00376031  0.03332671  0.0580061   0.00414042
  0.02169008  0.02133783  0.01015558  0.00566857  0.01414039  0.00777258
  0.0318885   0.          0.05904908  0.0241606   0.00260791  0.          0.
  0.          0.          0.01060661  0.02383722  0.          0.00400762
  0.00041497  0.01207875  0.01779926  0.00761597  0.03944514  0.03165292
  0.00208581  0.02744033  0.01668677  0.03028724  0.00660357  0.02400281
  0.01587162  0.07190543  0.07313231  0.01682512  0.05859096  0.00279908
  0.00605451  0.02980045  0.09671787  0.          0.          0.
  0.00036551  0.01100901  0.          0.          0.        ]
```

In [15]:

```
data.shape
```

Out[15]:

```
(251889, 19)
```

In [25]:

```
train_02.columns
```

Out[25]:

```
Index(['discount_rate', 'distance', 'day_of_month', 'days_distance',
       'discount_man', 'discount_jian', 'is_man_jian', 'total_sales',
       'sales_use_coupon', 'total_coupon', 'merchant_min_distance',
       'merchant_max_distance', 'merchant_mean_distance',
       'merchant_median_distance', 'merchant_coupon_transfer_rate',
       'coupon_rate', 'count_merchant', 'user_min_distance',
       'user_max_distance', 'user_mean_distance', 'user_median_distance',
       'buy_use_coupon', 'buy_total', 'coupon_received',
       'avg_user_date_datereceived_gap', 'min_user_date_datereceived_gap',
       'max_user_date_datereceived_gap', 'buy_use_coupon_rate',
       'user_coupon_transfer_rate', 'user_merchant_buy_total',
       'user_merchant_received', 'user_merchant_buy_use_coupon',
       'user_merchant_any', 'user_merchant_buy_common',
       'user_merchant_coupon_transfer_rate', 'user_merchant_coupon_buy_rate',
       'user_merchant_rate', 'user_merchant_common_buy_rate',
       'this_month_user_receive_same_coupon_count',
       'this_month_user_receive_all_coupon_count',
       'this_month_user_receive_same_coupon_lastone',
       'this_month_user_receive_same_coupon_firstone',
       'this_day_user_receive_all_coupon_count',
       'this_day_user_receive_same_coupon_count', 'day_gap_before',
       'day_gap_after', 'is_weekend', 'weekday1', 'weekday2', 'weekday3',
       'weekday4', 'weekday5', 'weekday6', 'weekday7', 'label'],
      dtype='object')
```

In [34]:

```
features = {}
for idx,col in enumerate(selector.estimator_.feature_importances_):
    features[train_02.columns[idx]] = col
```

In [38]:

```
print(features)
```

```
{'total_sales': 0.021337831566085888, 'day_gap_before': 0.029800445403043729, 'day_o
f_month': 0.00376031031780909, 'user_coupon_transfer_rate': 0.0076159658867916398,
'user_median_distance': 0.0, 'weekday2': 0.0, 'user_merchant_common_buy_rate': 0.015
871615063068646, 'weekday6': 0.0, 'merchant_mean_distance': 0.031888496179852607, 'b
uy_total': 0.023837221011506524, 'this_day_user_receive_same_coupon_count': 0.006054
5055342317621, 'total_coupon': 0.0056685737910950893, 'merchant_max_distance': 0.007
7725798867157424, 'coupon_rate': 0.024160599432051538, 'merchant_min_distance': 0.01
4140388639626916, 'weekday4': 0.011009013978989656, 'distance': 0.01465190998527880
1, 'user_max_distance': 0.0, 'discount_rate': 0.050006578382151251, 'weekday5': 0.0,
'this_month_user_receive_same_coupon_lastone': 0.016825118539721514, 'buy_use_coupon
_rate': 0.017799256923501244, 'days_distance': 0.033326713436473379, 'this_day_user_
receive_all_coupon_count': 0.0027990817831094802, 'avg_user_date_datereceived_gap':
0.0040076221072483558, 'user_merchant_rate': 0.024002806420404151, 'weekday1': 0.0,
'discount_jian': 0.0041404168991677195, 'user_merchant_coupon_transfer_rate': 0.0302
8724088863485, 'weekday3': 0.00036551137272462027, 'sales_use_coupon': 0.01015558029
8906078, 'min_user_date_datereceived_gap': 0.00041497461570457752, 'this_month_user_
receive_same_coupon_firstone': 0.058590962936327887, 'user_merchant_buy_total': 0.03
9445144948616992, 'count_merchant': 0.0026079096092790238, 'weekday7': 0.0, 'discoun
t_man': 0.058006095923711552, 'is_weekend': 0.0, 'user_merchant_buy_use_coupon': 0.0
0208580869845960002, 'merchant_coupon_transfer_rate': 0.059049075013072633, 'coupon_r
eceived': 0.0, 'day_gap_after': 0.096717873028341311, 'user_merchant_any': 0.0274403
26697185453, 'user_merchant_buy_common': 0.016686773966436211, 'user_merchant_coupon
_buy_rate': 0.0066035653012397402, 'user_min_distance': 0.0, 'user_merchant_receive
d': 0.031652915972068964, 'buy_use_coupon': 0.010606614582289630, 'is_man_jian': 0.02
1690084024321547, 'this_month_user_receive_all_coupon_count': 0.073132307366944233,
'max_user_date_datereceived_gap': 0.012078752568954989, 'user_mean_distance': 0.0,
'merchant_median_distance': 0.0, 'this_month_user_receive_same_coupon_count': 0.0719
05431018855395}
```

In [ ]:

```

```

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from numpy import vstack, array, nan
from sklearn.datasets import load_iris
from sklearn import preprocessing
from sklearn import feature_selection
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
features_new = [ 'day_gap_after','this_month_user_receive_all_coupon_count', 'this_month_user_receiv

print("train01")
train01_data = pd.read_csv('train01.csv')
train01_data_new = train01_data[features_new]
train01_data_new_np = train01_data_new.values
imp = preprocessing.Imputer(missing_values='NaN', strategy='mean', axis=0)
train01_data_new = imp.fit_transform(train01_data_new_np)
train01_data_new_df = pd.DataFrame(train01_data_new, columns=features_new)
train01_data_new_df_1 = train01_data_new_df[train01_data_new_df.label >-1]
train01_data_new_df_1_np = train01_data_new_df_1.values
print(train01_data_new_df_1.info())


print("train02")
train02_data = pd.read_csv('train02.csv')
train02_data_new = train02_data[features_new]
train02_data_new_np = train02_data_new.values
imp = preprocessing.Imputer(missing_values='NaN', strategy='mean', axis=0)
train02_data_new = imp.fit_transform(train02_data_new_np)
train02_data_new_df = pd.DataFrame(train02_data_new, columns=features_new)
train02_data_new_df_2 = train02_data_new_df[train02_data_new_df.label >-1]
train02_data_new_df_2_np = train02_data_new_df_2.values
print(train02_data_new_df_2.info())

train_data_new_np = np.concatenate([train01_data_new_df_1,train02_data_new_df_2],axis = 0)

train_data_new_df = pd.DataFrame(train_data_new_np, columns=features_new)
print(train_data_new_df.info())
train_data_new_df.to_csv("processed_data.csv")
```

```
train01
<class 'pandas.core.frame.DataFrame'>
Int64Index: 486936 entries, 0 to 492695
Data columns (total 20 columns):
day_gap_after                                 486936 non-null float64
this_month_user_receive_all_coupon_count      486936 non-null float64
this_month_user_receive_same_coupon_count     486936 non-null float64
merchant_coupon_transfer_rate                 486936 non-null float64
this_month_user_receive_same_coupon_firstone  486936 non-null float64
discount_man                                  486936 non-null float64
discount_rate                                 486936 non-null float64
user_merchant_buy_total                       486936 non-null float64
days_distance                                 486936 non-null float64
```

```
merchant_mean_distance                      486936 non-null float64
user_merchant_received                      486936 non-null float64
user_merchant_coupon_transfer_rate          486936 non-null float64
day_gap_before                              486936 non-null float64
user_merchant_any                           486936 non-null float64
coupon_rate                                 486936 non-null float64
user_merchant_rate                          486936 non-null float64
buy_total                                   486936 non-null float64
is_man_jian                                 486936 non-null float64
total_sales                                 486936 non-null float64
label                                       486936 non-null float64
dtypes: float64(20)
memory usage: 78.0 MB
None
train02
<class 'pandas.core.frame.DataFrame'>
Int64Index: 251889 entries, 0 to 255223
Data columns (total 20 columns):
day_gap_after                               251889 non-null float64
this_month_user_receive_all_coupon_count    251889 non-null float64
this_month_user_receive_same_coupon_count   251889 non-null float64
merchant_coupon_transfer_rate               251889 non-null float64
this_month_user_receive_same_coupon_firstone 251889 non-null float64
discount_man                                251889 non-null float64
discount_rate                               251889 non-null float64
user_merchant_buy_total                     251889 non-null float64
days_distance                               251889 non-null float64
merchant_mean_distance                      251889 non-null float64
user_merchant_received                      251889 non-null float64
user_merchant_coupon_transfer_rate          251889 non-null float64
day_gap_before                              251889 non-null float64
user_merchant_any                           251889 non-null float64
coupon_rate                                 251889 non-null float64
user_merchant_rate                          251889 non-null float64
buy_total                                   251889 non-null float64
is_man_jian                                 251889 non-null float64
total_sales                                 251889 non-null float64
label                                       251889 non-null float64
dtypes: float64(20)
memory usage: 40.4 MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 738825 entries, 0 to 738824
Data columns (total 20 columns):
day_gap_after                               738825 non-null float64
this_month_user_receive_all_coupon_count    738825 non-null float64
this_month_user_receive_same_coupon_count   738825 non-null float64
merchant_coupon_transfer_rate               738825 non-null float64
this_month_user_receive_same_coupon_firstone 738825 non-null float64
discount_man                                738825 non-null float64
discount_rate                               738825 non-null float64
user_merchant_buy_total                     738825 non-null float64
days_distance                               738825 non-null float64
merchant_mean_distance                      738825 non-null float64
user_merchant_received                      738825 non-null float64
user_merchant_coupon_transfer_rate          738825 non-null float64
day_gap_before                              738825 non-null float64
user_merchant_any                           738825 non-null float64
coupon_rate                                 738825 non-null float64
user_merchant_rate                          738825 non-null float64
buy_total                                   738825 non-null float64
```

```
is_man_jian                738825 non-null float64
total_sales                738825 non-null float64
label                      738825 non-null float64
dtypes: float64(20)
memory usage: 112.7 MB
None
```

In [12]:

```
# 模型结果
import sys
import io
import numpy as np
import matplotlib.pyplot as plt
from sklearn.learning_curve import learning_curve
from sklearn import linear_model
from sklearn.ensemble import RandomForestRegressor
import pandas as pd #数据分析
from sklearn import linear_model
from sklearn.ensemble import RandomForestClassifier

train01_data = pd.read_csv('processed_data.csv', header=0)

from sklearn import preprocessing

#print("train01")
features = list(train01_data.columns)

imp = preprocessing.Imputer(missing_values='NaN', strategy='mean', axis=0)
train01_data_new = imp.fit_transform(train01_data)
train01_data_new_df = pd.DataFrame(train01_data_new, columns=features)
train01_data_new_df_1 = train01_data_new_df[train01_data_new_df.label >-1]
#print(train01_data_new_df_1.info())

train_np = train01_data_new_df_1.as_matrix()

# y即Survival结果
y = train_np[:, -1]
# X即特征属性值
X = train_np[:, :-1]

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.1,
    random_state=42
)
#print("X")
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import Normalizer
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import GradientBoostingClassifier

# (5) 模型构建与训练
# clf = linear_model.LogisticRegression(C=100.0, penalty='l1', tol=1e-6)
clf = RandomForestClassifier(criterion='gini', max_depth=5, n_estimators=5)

from sklearn.metrics import classification_report
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
             precision    recall  f1-score   support

        0.0       0.94      1.00      0.97     23664
        1.0       0.75      0.08      0.14      1525
```

avg / total	0.93	0.94	0.92	25189

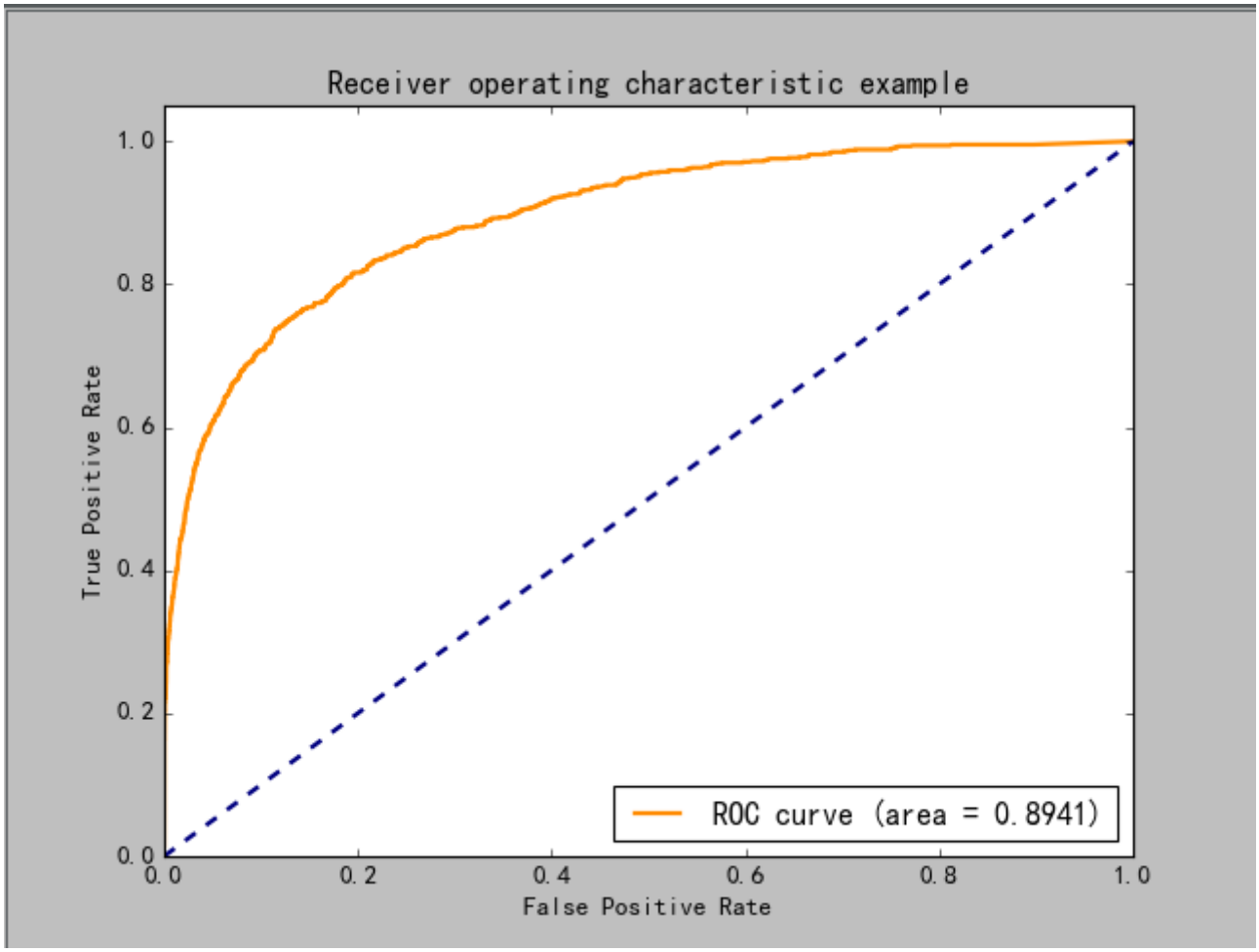
Type *Markdown* and LaTeX: α^2

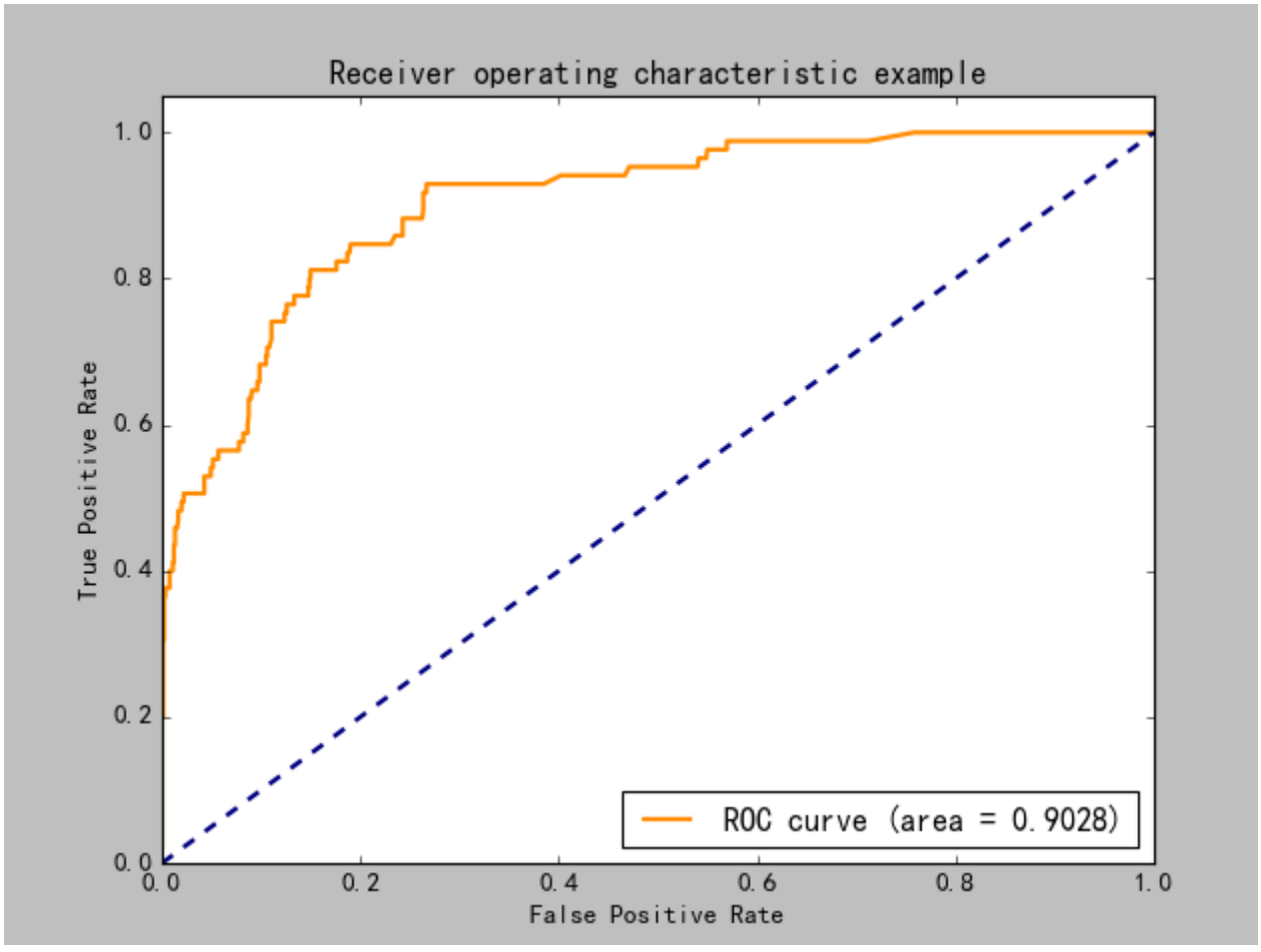
```
Grid scores on development set:

0.876 (+/-0.013) for {'n_estimators': 10, 'min_samples_leaf': 2, 'criterion': 'gini'}
0.887 (+/-0.013) for {'n_estimators': 15, 'min_samples_leaf': 2, 'criterion': 'gini'}
0.891 (+/-0.014) for {'n_estimators': 20, 'min_samples_leaf': 2, 'criterion': 'gini'}
0.883 (+/-0.008) for {'n_estimators': 10, 'min_samples_leaf': 4, 'criterion': 'gini'}
0.889 (+/-0.008) for {'n_estimators': 15, 'min_samples_leaf': 4, 'criterion': 'gini'}
0.892 (+/-0.009) for {'n_estimators': 20, 'min_samples_leaf': 4, 'criterion': 'gini'}
0.885 (+/-0.010) for {'n_estimators': 10, 'min_samples_leaf': 6, 'criterion': 'gini'}
0.889 (+/-0.010) for {'n_estimators': 15, 'min_samples_leaf': 6, 'criterion': 'gini'}
0.891 (+/-0.010) for {'n_estimators': 20, 'min_samples_leaf': 6, 'criterion': 'gini'}
0.877 (+/-0.012) for {'n_estimators': 10, 'min_samples_leaf': 2, 'criterion': 'entropy'}
0.887 (+/-0.011) for {'n_estimators': 15, 'min_samples_leaf': 2, 'criterion': 'entropy'}
0.893 (+/-0.013) for {'n_estimators': 20, 'min_samples_leaf': 2, 'criterion': 'entropy'}
0.882 (+/-0.007) for {'n_estimators': 10, 'min_samples_leaf': 4, 'criterion': 'entropy'}
0.890 (+/-0.010) for {'n_estimators': 15, 'min_samples_leaf': 4, 'criterion': 'entropy'}
0.894 (+/-0.011) for {'n_estimators': 20, 'min_samples_leaf': 4, 'criterion': 'entropy'}
0.884 (+/-0.008) for {'n_estimators': 10, 'min_samples_leaf': 6, 'criterion': 'entropy'}
0.889 (+/-0.009) for {'n_estimators': 15, 'min_samples_leaf': 6, 'criterion': 'entropy'}
0.892 (+/-0.009) for {'n_estimators': 20, 'min_samples_leaf': 6, 'criterion': 'entropy'}
```

```
Best parameters set found on development set:

{'n_estimators': 20, 'min_samples_leaf': 4, 'criterion': 'entropy'}
```


Receiver operating characteristic example
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.8941)


Receiver operating characteristic example
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.9028)

In [ ]:

```

```