group2

任务4

渠道类型分析

任务4：渠道类型分析

数据预处理

- **去除空值**
- **去除无意义的值**
  - 去除性别为“无法区分”
  - 去除终端类型为‘0’
  - 去除客户类型不为‘公共客户’的类型
  - 去除渠道类型描述为‘其他类型的’

- **数据整理——调整格式与赋值**
- 首先对数据表中的数字字符数据进行格式转换，转换为整数或浮点数类型；
- 将文本变量进行编码

如性别编码为{‘男’：1，‘女’：0}，终端类型编码{‘2G’:1,‘3G’:2,‘4G’:3}等。

- 去除‘产品分类’变量

Baseline 选择

- **以全部预测为‘社会渠道’为baseline**
- 准确度为 54%

```
#baseline 1: 全部预测为社会渠道
acc=trial2.groupby(['channel'])['用户ID'].count()[0]/len(trial2)
#0.5424743072720177
print('全部预测为社会渠道:acc=',acc)
```

```
全部预测为社会渠道:acc= 0.5424743072720177
```

任务4：渠道类型分析

逻辑回归

- **初始模型**
- 全部为默认参数

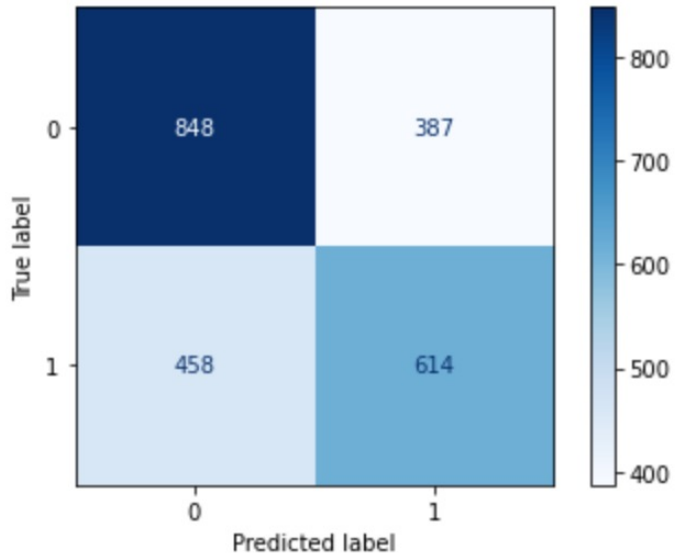
```
confusion matrix:
 preds     0    1
actual
0        857  378
1        499  573
accuracy= 0.6198526224534027
precision= 0.6939271255060728
recall= 0.6320058997050148
```

- **优化方法1：数据变换**
- 将数据标准化处理

```
confusion matrix:
 preds     0    1
actual
0        856  379
1        474  598
accuracy= 0.6302557433896836
precision= 0.6931174089068826
recall= 0.643609022556391
```

- **优化方法2：特征选择**
- 通过RFECV进行特征降维

```
accuracy= 0.6337234503684439
precision= 0.6866396761133603
recall= 0.6493108728943339
```



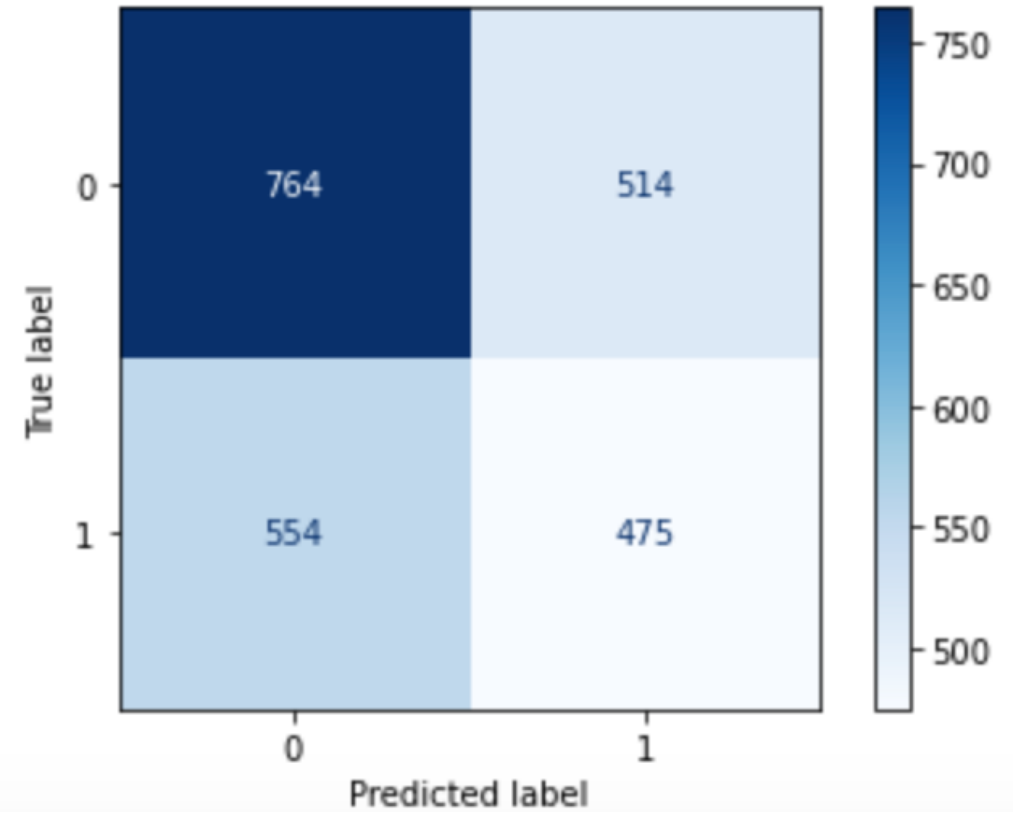
任务4：渠道类型分析

KNN

- **初始模型**
- K的初始值设置为3

```
confusion matrix:
 preds      0    1
actual
0         764  514
1         554  475
accuracy= 0.5370611183355006
precision= 0.5978090766823161
recall= 0.5796661608497724
```

- 拟合结果较差，对模型进行优化



任务4：渠道类型分析

KNN

- **优化方法1：数据变换**
- 考虑三种方式：
  - 不做变换
  - 标准化处理
  - 按最值区间缩放
- 最终选择将特征按照最值区间缩放，

  发现模型准确度有显著提升

```
X_train, X_test, y_train, y_test =
    train_test_split(data, target, test_size=0.3, random_state=50)
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train,y_train)
y_pred=knn.predict(X_test)
print('original accuracy= ',accuracy_score(y_test, y_pred))

s = StandardScaler()
s_data = s.fit_transform(data)
X_train, X_test, y_train, y_test =
    train_test_split(s_data, target, test_size=0.3, random_state=100)
knn.fit(X_train,y_train)
y_pred=knn.predict(X_test)
print('Standard accuracy= ',accuracy_score(y_test, y_pred))

mm = MinMaxScaler()
m_data = mm.fit_transform(data)
X_train, X_test, y_train, y_test =
    train_test_split(m_data, target, test_size=0.3, random_state=100)
knn.fit(X_train,y_train)
y_pred=knn.predict(X_test)
print('MinMax accuracy= ',accuracy_score(y_test, y_pred))
```

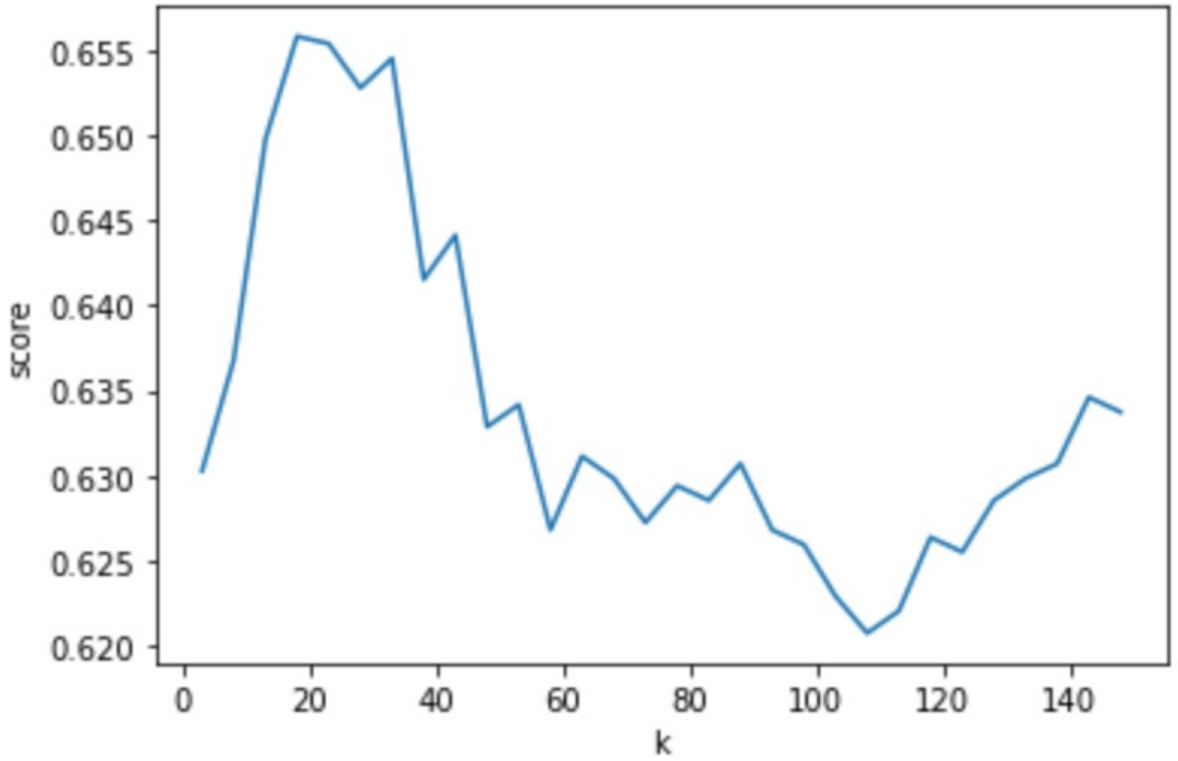
```
original accuracy=  0.5370611183355006
Standard accuracy=  0.6168183788469874
MinMax accuracy=  0.6302557433896836
```

任务4：渠道类型分析

KNN

- **优化方法2: 参数优化**
- 通过学习曲线，寻找最优参数 k
- 得到最优参数 k = 18

```
best k =  18
```

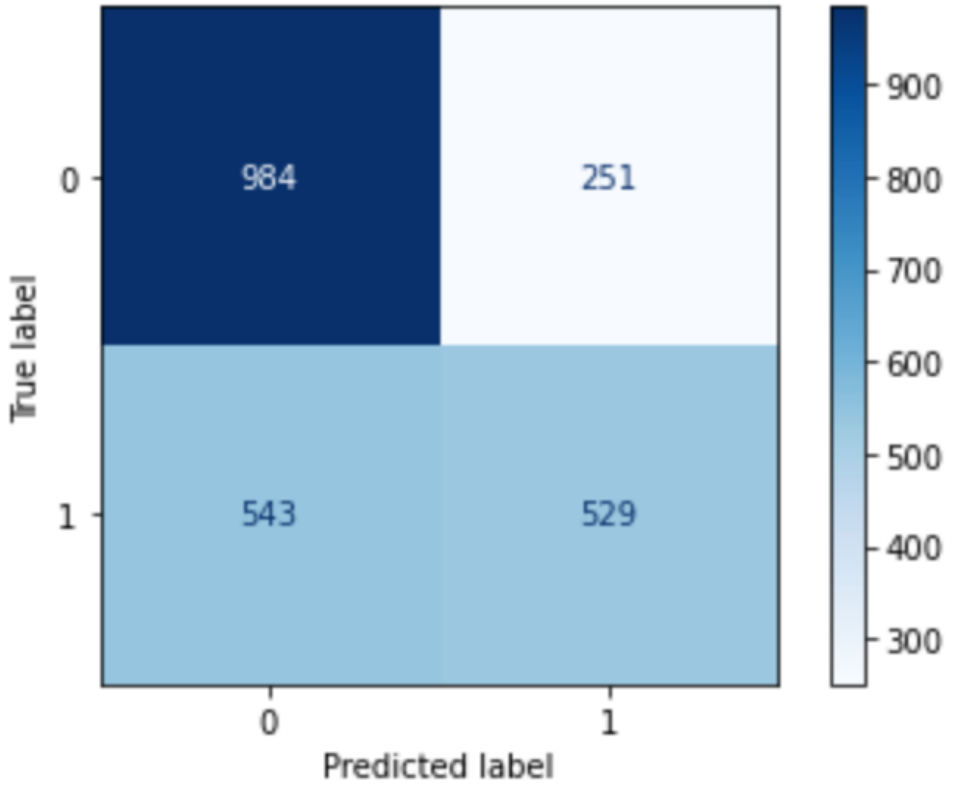


任务4：渠道类型分析

KNN

- **最优模型**
- 准确度为 66%
- 查准率为 80%
- 查全率为 64%

```
confusion matrix:
 preds      0    1
actual
0         984  251
1         543  529
accuracy= 0.6558300823580407
precision= 0.7967611336032389
recall= 0.6444007858546169
```



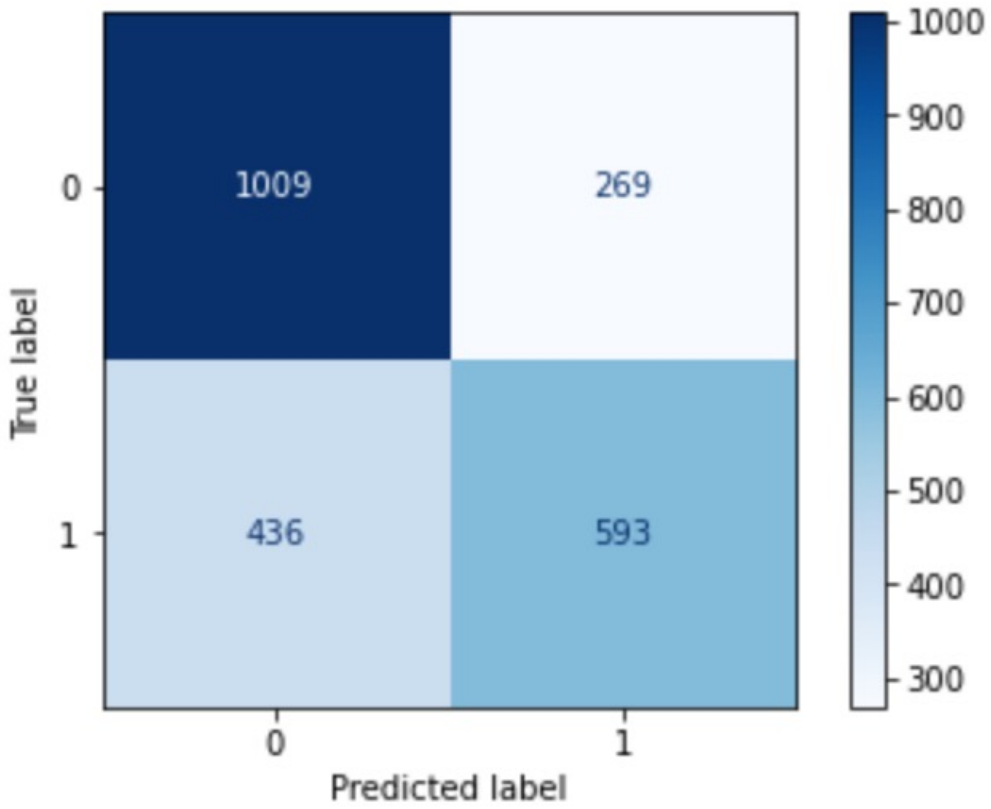
任务4：渠道类型分析

随机森林

- **初始模型**
- 所有参数皆为默认值

```
confusion matrix:
 preds       0    1
actual
0         1009  269
1          436  593
accuracy= 0.694408322496749
precision= 0.7895148669796557
recall= 0.6982698961937717
cross_val_score= 0.7063878174512862
obb= 0.6966542750929368
```

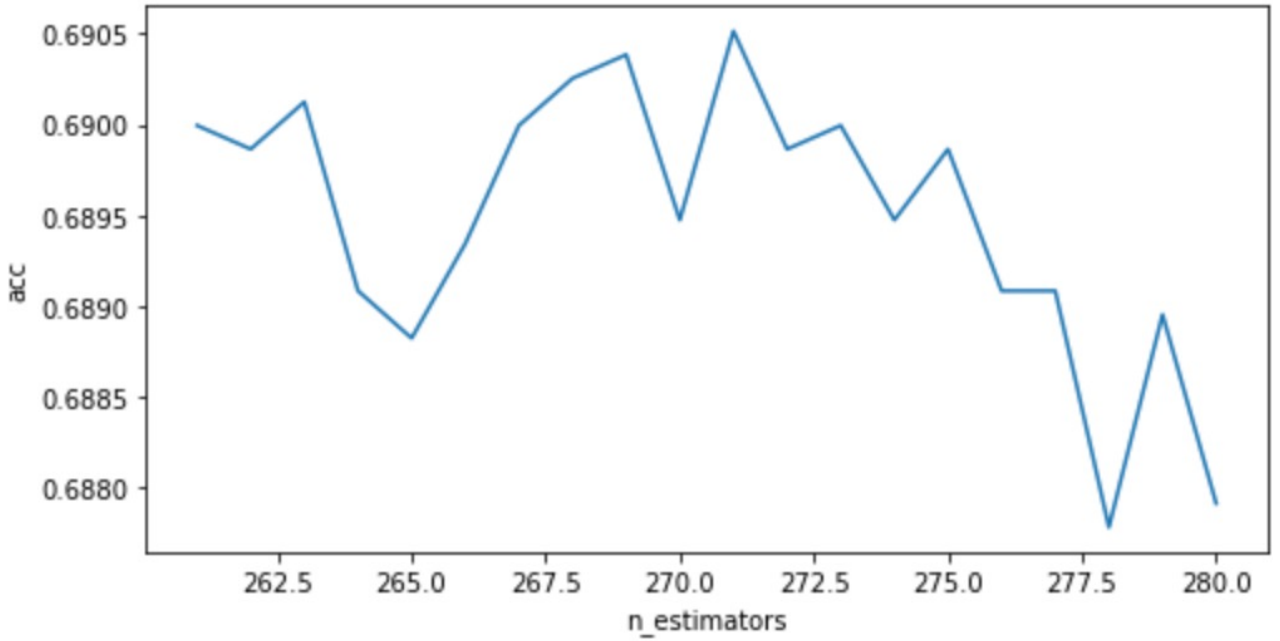
- 拟合效果较KNN更好



任务4：渠道类型分析

随机森林

- **优化方法: 超参优化**
- 由于数据变换，对于随机森林模型的准确度提升不大，所以不考虑对数据进行变换
- 采用方法：
- 利用cross_val_score完成 K折交叉验证
- **优化参数1: n_estimators(森林中树的数量)**
- 通过学习曲线，寻找最优 cross_val_score




```
271 0.690515612718949
```

任务4：渠道类型分析

随机森林

- **优化方法: 超参优化**
- **优化参数2: n_ features(特征的数量)**
- **优化参数3: max_depth（树的最大深度）**
- 采用方法：
- 通过网格搜索寻找最优

cross_val_score

```
param_grid = {'max_features' : np.arange(1,17,1)}
#一般根据数据大小进行尝试，像该数据集 可从1-10 或1-20开始
rf = RandomForestClassifier(n_estimators=n_est,random_state=50,n_jobs=-1
GS = GridSearchCV(rf,param_grid,cv=5)
GS.fit(data,target)
max_f=GS.best_params_['max_features']
print(GS.best_params_)
print(GS.best_score_)
```

```
{'max_features': 12}
0.7009244022393445
```

```
param_grid = {'max_depth' : np.arange(1,30,1)}
#一般根据数据大小进行尝试，像该数据集 可从1-10 或1-20开始
rf = RandomForestClassifier(n_estimators=n_est,max_features=max_f,random
GS = GridSearchCV(rf,param_grid,cv=5)
GS.fit(data,target)
max_d=GS.best_params_['max_depth']
print(GS.best_params_)
print(GS.best_score_)
```

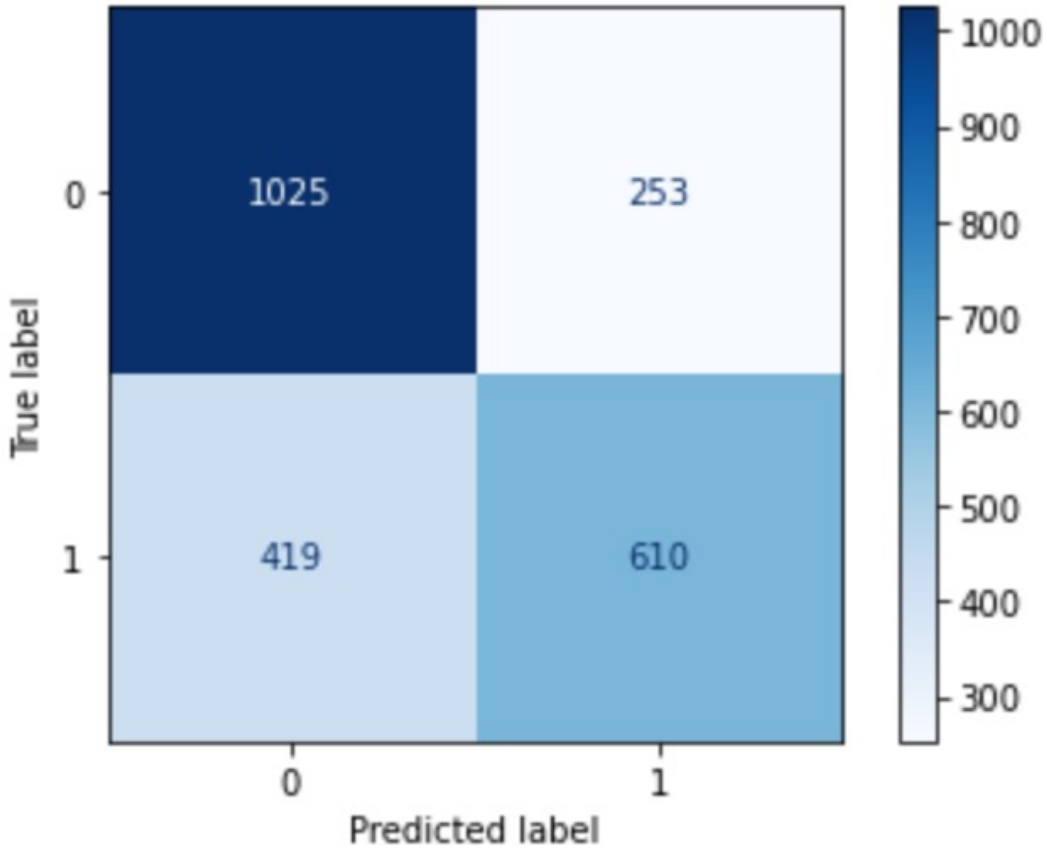
```
{'max_depth': 13}
0.7074272834875838
```

任务4：渠道类型分析

随机森林

- **最优模型**
- 将搜索到的最优参数带入模型
- 准确度为 71%
- 查准率为 80%
- 查全率为 71%

```
confusion matrix:
 preds       0    1
actual
0         1025  253
1          419  610
accuracy= 0.7087126137841352
precision= 0.8020344287949922
recall= 0.7098337950138505
cross_val_score= 0.7063878174512862
obb= 0.6966542750929368
```



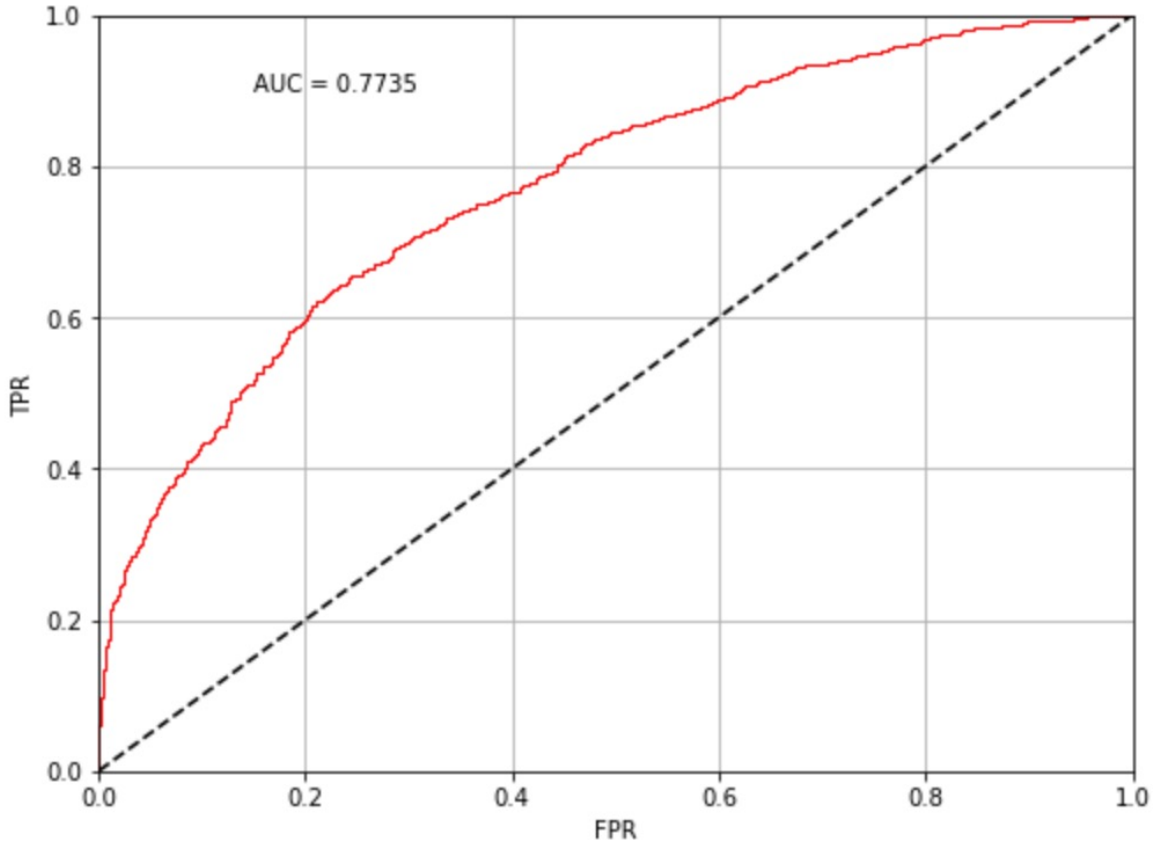
任务4：渠道类型分析

随机森林

- **最优模型**
- 将搜索到的最优参数带入模型
- 准确度为 71%
- 查准率为 80%
- 查全率为 71%

```
confusion matrix:
 preds       0    1
actual
0        1025  253
1         419  610
accuracy= 0.7087126137841352
precision= 0.8020344287949922
recall= 0.7098337950138505
cross_val_score= 0.7063878174512862
obb= 0.6966542750929368
```

- **ROC曲线：**



任务4：渠道类型分析

随机森林

- **最优模型**
- 将搜索到的最优参数带入模型
- 准确度为 71%
- 查准率为 80%
- 查全率为 71%

```
confusion matrix:
 preds       0    1
actual
0         1025  253
1          419  610
accuracy= 0.7087126137841352
precision= 0.8020344287949922
recall= 0.7098337950138505
cross_val_score= 0.7063878174512862
obb= 0.6966542750929368
```

- **特征重要性排序：**

1）	性别	0.146783
2）	年龄	0.138733
3）	归属地	0.089952
4）	在网时长	0.079088
5）	换机频率	0.076729
6）	终端类型	0.070941
7）	最近使用操作系统偏好	0.070757
8）	上网流量使用	0.059099
9）	漫游流量使用	0.057437
10）	总收入	0.051996
11）	增值收入	0.051297
12）	流量收入	0.042586
13）	短信收入	0.037483
14）	彩信收入	0.010720
15）	语音收入	0.008336
16）	是否欠费	0.006476
17）	产品大类	0.001588