机器学习期末课程论文

——监督学习与半监督学习对比

一、任务背景与数据集介绍

本实验数据集来自哥伦布亚商学院教授 Ray Fisman 和 Sheena Iyengar 在 2002 年至 2004 年中做的实验，他们邀请志愿者参加闪电速配实验(相亲车轮战，每 4 分钟与一名相亲对象快速沟通，然后再换下一个相亲对象)。

在实验中使用阿里云 DSW 平台，使用

```
!wget https://pai-public-data.oss-cn-beijing.aliyuncs.com/speed_dating/Speed%20Dating%20Data%20Key.doc
```

表示数据信息

```
!wget https://pai-public-data.oss-cn-beijing.aliyuncs.com/speed_dating/Speed%20Dating%20Data.csv
```

导入数据集

并附上上述文件的.CSV 版在压缩包中。

实验提供了一些相关的个人信息给相亲对象，并询问相亲对象给出是否愿意在不久的未来再次见面。如果某次相亲的双方都选择愿意与对方之后再见面，则双方的数据项中 match 均为 1,否则只要有一方不愿意 match 均为 0.

数据集(详细的数据介绍附在 Speed Dating Data Key.doc 文件中，这里介绍一些较为重要的数据项)包括参与人员的 id，参与的场次，性别，年龄，SAT 分数，种族，院系，学校，attr--attractive 吸引力指数，sinc--sincere 真诚度指数，intel--intelligent 聪明指数，fun--有趣指数，amb--ambitious 雄心指数，shar--has shared interests/hobbies--有同样兴趣爱好的程度，等等 195 个特征、8277 条带标签数据。

其中对于以上六种指数加上不同的后缀表示被试者在相亲前、相亲后当晚、相亲后一段时间后对于对方在上述指标上所期待/所感受到的程度以及对自己的这些指标的评价。

而我们在后面预测中使用到的 attr_o,sinc_o,intel_o,fun_o,amb_o, share_o,则表示当天晚上闪电速配结束之后被试者给其同伴在上述指标上打的分数。

二、所采取的方法

首先利用数据可视化工具-饼图、柱形图对数据有一个基本掌控，并且了解到数据缺失值的分布情况，然后将变量相关关系进行可视化得到与 match 相关关系最大的几个变量，在这里由于我们在数据可视化时发现 match 仅与有限个变量有明显的相关关系，所以直接将其加入模型中进行预测而不再进一步做变量选择。

利用 SMOTE 合成少数类过采样技术解决数据不均衡问题。

监督学习使用了五种模型-随机森林、逻辑回归、BaggingClassifier、AdaBoost、支持向量机；半监督学习则对于上述五种模型中的前四种做 self-training 对比。

三、实验

（一）数据 EDA

1. 缺失值了解

	column_name	percent_missing
iid	iid	0.000000
dec_o	dec_o	0.000000
samerace	samerace	0.000000
match	match	0.000000
partner	partner	0.000000
...	...	...
amb7_2	amb7_2	76.665075
sinc7_2	sinc7_2	76.665075
expnum	expnum	78.515159
numdat_3	numdat_3	82.143710
num_in_3	num_in_3	92.026737

195 rows × 2 columns

图 1 缺失值了解

发现缺失值很多，所以下一步要进行变量的选择，否则将缺失值很多的变量

加入模型会造成过拟合。

2. 闪电速配成功比例

1) 所有人中在本次闪电速配中找到对象与未成功比例

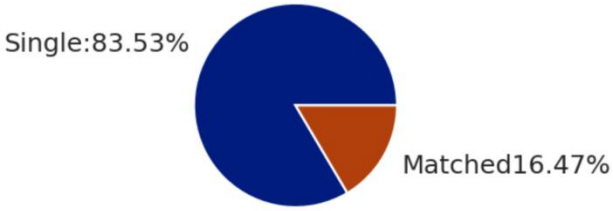


图 2 所有人中在本次闪电速配中找到对象与未成功比例

2) 女生中在本次闪电速配中找到对象与未成功比例

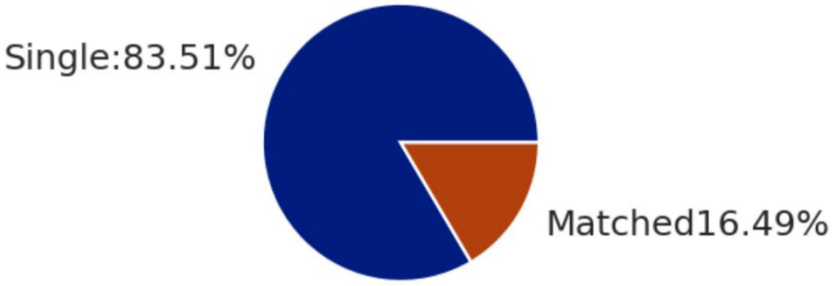


图 3 女生中在本次闪电速配中找到对象与未成功比例

3) 男生中在本次闪电速配中找到对象与未成功比例

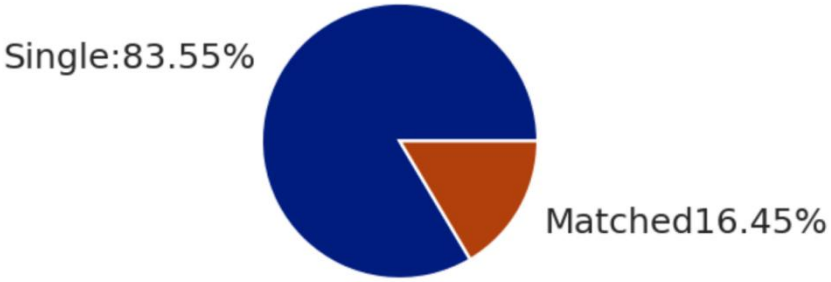


图 4 男生中在本次闪电速配中找到对象与未成功比例

从中我们发现本次闪电速配的成功率平均在六分之一，也初步表现出本次数据正例比反例要少得多的特性，数据十分不均衡。

3. 参与者年龄分布

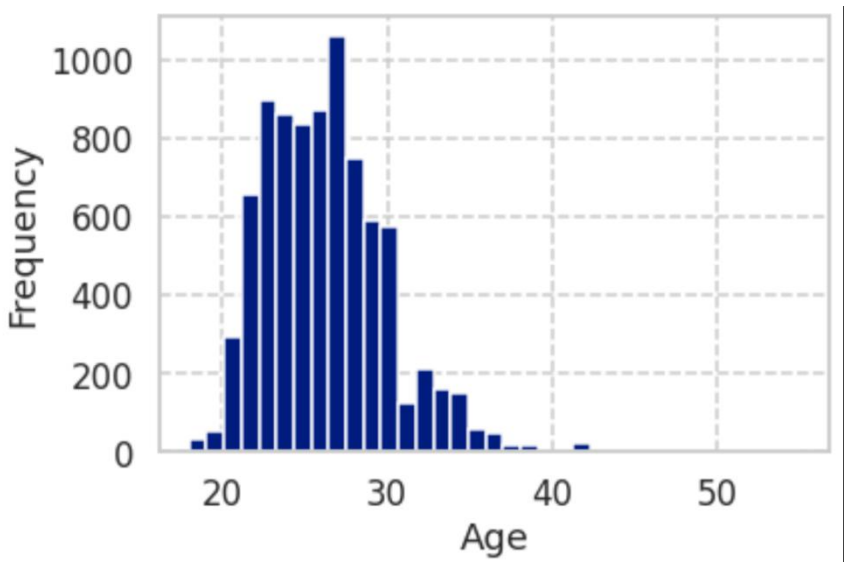


图 5 参与者年龄分布

发现参与者年龄集中在 21-30 岁之间。

4. 变量相关性

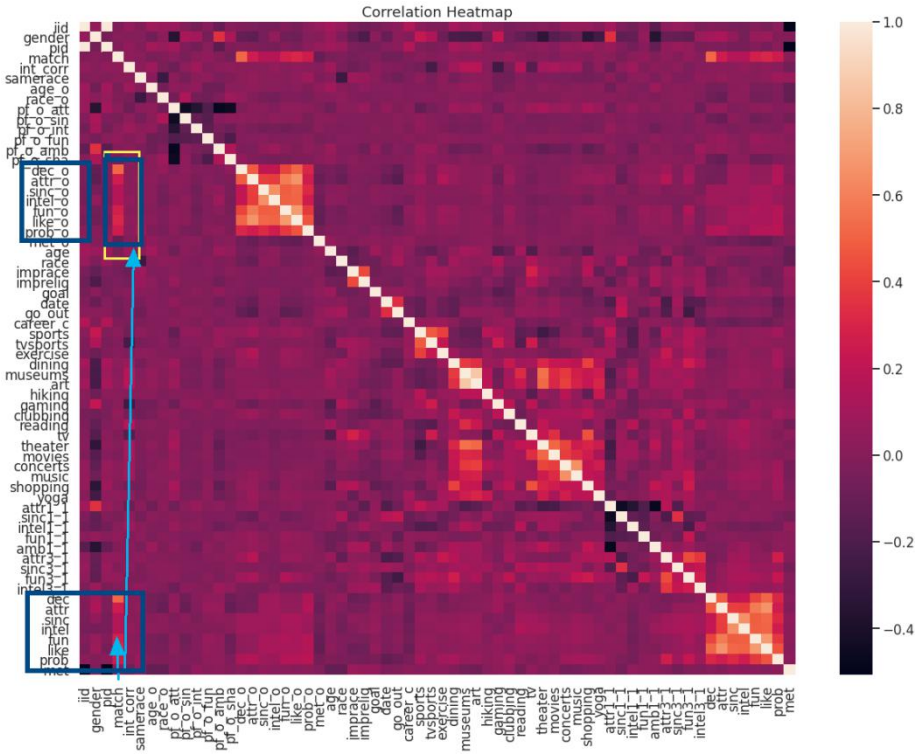


图 6 相关性图

发现 attr_o,sinc_o,intel_o,fun_o,amb_o,share_o 较其它变量相比与 match 有明显的相关性。且这些变量之间也有明显的相关性，因此我们选择这些变量加入模型进行预测。

（二）数据预处理

首先将上述六个特征 attr_o,sinc_o,intel_o,fun_o,amb_o,share_o 及预测变量 match 提取出来，去掉缺失值，得到 7031×7 条数据，命名为 clean_df.

clean_df	attr_o	sinc_o	intel_o	fun_o	amb_o	shar_o	match
0	6.0	8.0	8.0	8.0	6.0	0	
1	7.0	8.0	10.0	7.0	7.0	5.0	0
2	10.0	10.0	10.0	10.0	10.0	10.0	1
3	7.0	8.0	9.0	8.0	9.0	8.0	1
4	8.0	7.0	9.0	6.0	9.0	7.0	1
...	...	...	...	...	...	...	...
7026	10.0	5.0	3.0	2.0	6.0	5.0	0
7027	6.0	3.0	7.0	3.0	7.0	2.0	0
7028	2.0	1.0	2.0	2.0	2.0	1.0	0
7029	5.0	7.0	5.0	5.0	3.0	6.0	0
7030	8.0	8.0	7.0	7.0	7.0	7.0	0

[7031 rows x 7 columns]

图 7 初步处理后的 clean_df

在上面的分析中我们发现该数据集的分布十分不均衡，所以我们要利用 SMOTE 对正类进行过采样，并且由于过采样之后新创造的数据均为正例且集中分布在后面，所以我们将数据集按行打乱，最后得到数据集 clean_df。

clean_df	attr_o	sinc_o	intel_o	fun_o	amb_o	shar_o	match
0	7.0	8.000000	6.000000	5.00000	8.0	4.000000	0
1	7.0	9.000000	8.000000	8.00000	6.0	5.628347	1
2	9.0	8.000000	7.000000	8.00000	7.0	6.275288	1
3	7.0	8.000000	8.000000	8.00000	7.0	8.000000	0
4	10.0	8.834164	9.000000	9.00000	9.0	9.000000	1
...	...	...	...	...	...	...	...
11631	8.0	8.000000	8.000000	8.00000	6.0	7.000000	1
11632	6.0	7.000000	7.000000	8.00000	7.0	5.000000	0
11633	8.0	9.328729	9.328729	9.00000	9.0	9.000000	1
11634	7.0	7.000000	8.000000	4.00000	7.0	7.000000	0
11635	5.0	5.000000	7.000000	7.94464	7.0	5.000000	1

[11636 rows x 7 columns]

图 8 最终得到的 clean_df

这里共有 11636 条数据，我们手动选取前 10000 条作为训练集，后 1636 条作为测试集。（由于我们在前面已经进行过数据的打乱了，所以这里手动选取也是可以的）

这 10000 条训练数据都是有标签的，为了后续做半监督学习，我们只选取前 2000 条保留其标签，用这 2000 条数据做监督学习，后 8000 条为无标签数据。

X_train_labeled	attr_o	sinc_o	intel_o	fun_o	amb_o	shar_o
0	8.000000	9.0	8.000000	7.0	7.000000	6.0
1	8.383216	8.0	8.383216	8.0	9.383216	7.0
2	8.000000	8.0	8.000000	6.0	4.000000	5.0
3	6.937606	7.0	7.000000	8.0	7.000000	6.0
4	7.000000	6.0	6.000000	5.0	5.000000	5.0
...	...	...	...	...	...	...
1995	7.000000	9.0	9.000000	8.0	6.000000	2.0
1996	5.000000	10.0	10.000000	6.0	10.000000	0.0
1997	10.000000	10.0	10.000000	10.0	10.000000	5.0
1998	9.000000	8.0	7.000000	7.0	7.000000	7.0
1999	6.000000	4.0	7.000000	7.0	8.000000	2.0

[2000 rows x 6 columns]

图 9 X_train_labeled：有标签数据特征值

X_train_unlabeled	attr_o	sinc_o	intel_o	fun_o	amb_o	shar_o
2000	6.000000	7.0	10.000000	9.0	10.000000	5.000000
2001	10.000000	10.0	10.000000	10.0	10.000000	10.000000
2002	6.000000	7.0	7.000000	6.0	7.000000	4.000000
2003	9.000000	9.0	7.000000	8.0	7.000000	6.000000
2004	7.000000	7.0	7.834458	7.0	5.000000	5.834458
...	...	...	...	...	...	...
9995	5.952815	5.0	7.047185	7.0	6.952815	6.000000
9996	7.000000	6.0	6.000000	6.0	6.000000	7.000000
9997	9.000000	8.0	8.000000	9.0	8.000000	9.000000
9998	6.000000	6.0	6.000000	5.0	4.000000	5.000000
9999	9.000000	9.0	9.000000	9.0	9.000000	9.000000

[8000 rows x 6 columns]

图 10 X_train_unlabeled：无标签数据特征值

y_train_labeled	0
1	1
2	1
3	1
4	0
..	
1995	0
1996	0
1997	1
1998	0
1999	0

Length: 2000, dtype: int64

图 11 y_train_labeled：有标签数据对应标签

（三）监督学习

发现首先利用监督学习的五个模型，利用上述六个变量对 match 做分类，先横向比较这五个模型本身的分类效果。

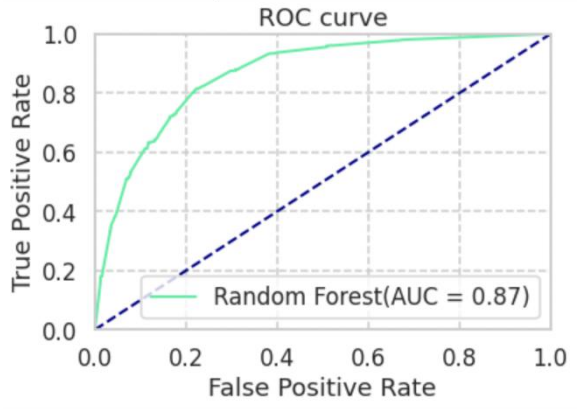
其中使用网格搜索对于每个模型找到最佳参数。

```
from sklearn.model_selection import GridSearchCV
def cross_validation_best_parameters(model, param_grid):
    grid = GridSearchCV(model, param_grid,cv=10, scoring='accuracy')
    X=clean_df.iloc[:,:-1].values
    y=clean_df.iloc[:,-1].values
    grid.fit(X,y)
    mean_scores = [result for result in grid.cv_results_['mean_test_score']]
    return mean_scores,grid.best_score_,grid.best_estimator_
logreg = linear_model.LogisticRegression(random_state=0)
c=[0.001, 0.01, 0.1, 1, 10, 100, 1000]
param_grid = dict(C=c)
mean_scores,Best_Accuracy, Best_classifier = cross_validation_best_parameters(logreg,param_grid)
print("Best accuracy is "+ str(Best_Accuracy))
print(Best_classifier)
```

```
Best accuracy is 0.6995531110347198
LogisticRegression(C=10, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=0, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

1. 随机森林：

```
Random Forest accuracy is  0.7799511002444988
```



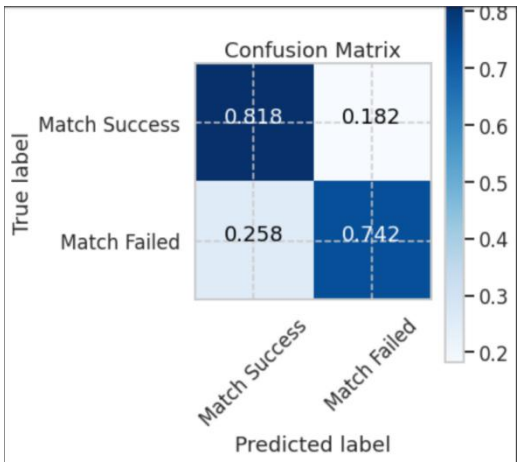


图 12 随机森林的 ROC 曲线和混淆矩阵

2. 逻辑回归

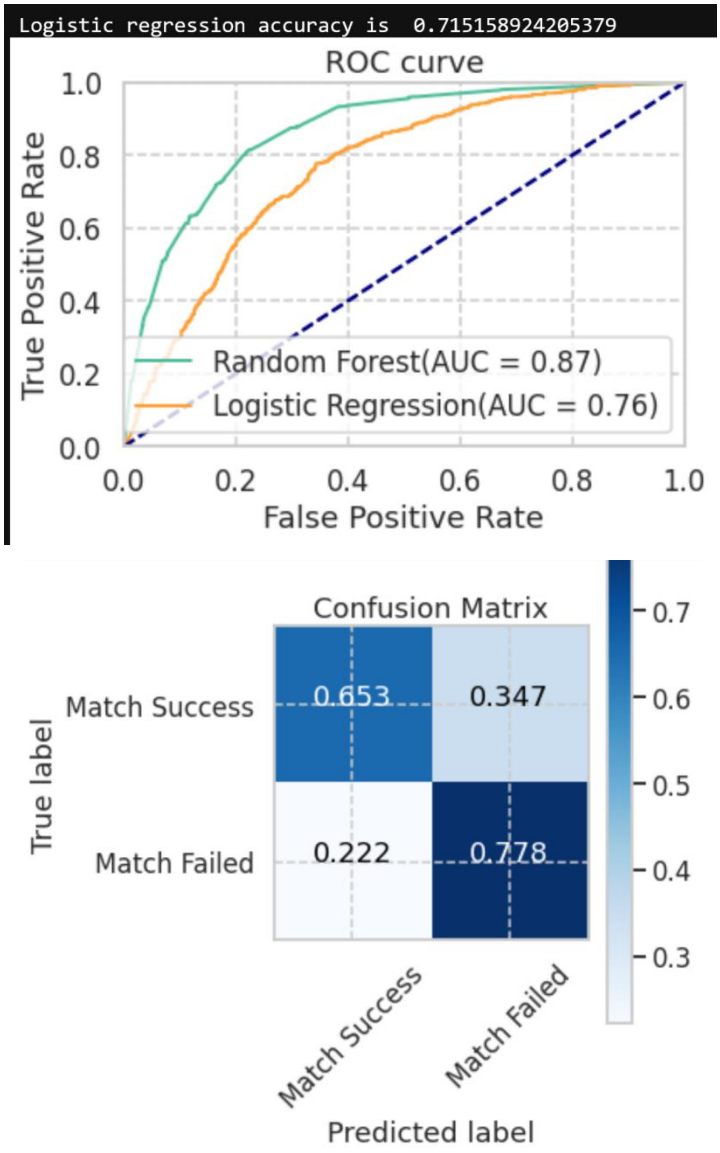
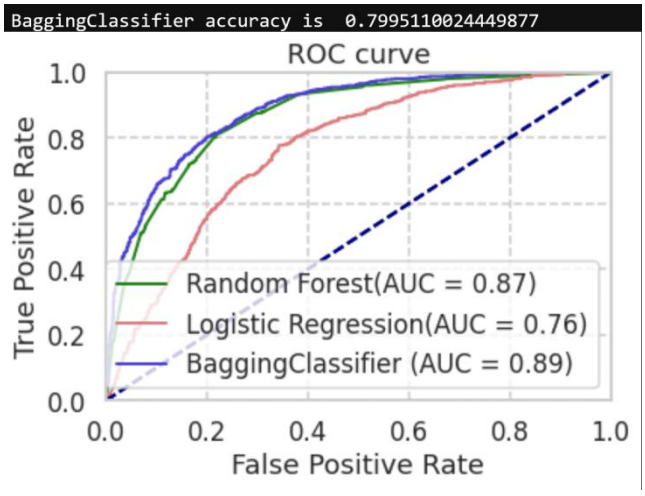


图 13　逻辑回归的 ROC 曲线和混淆矩阵

3. BaggingClassifier



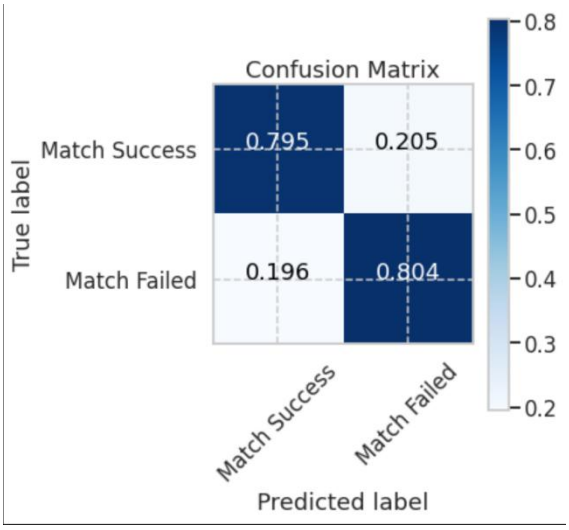


图 14　BaggingClassifier 的 ROC 曲线和混淆矩阵

4. AdaBoost

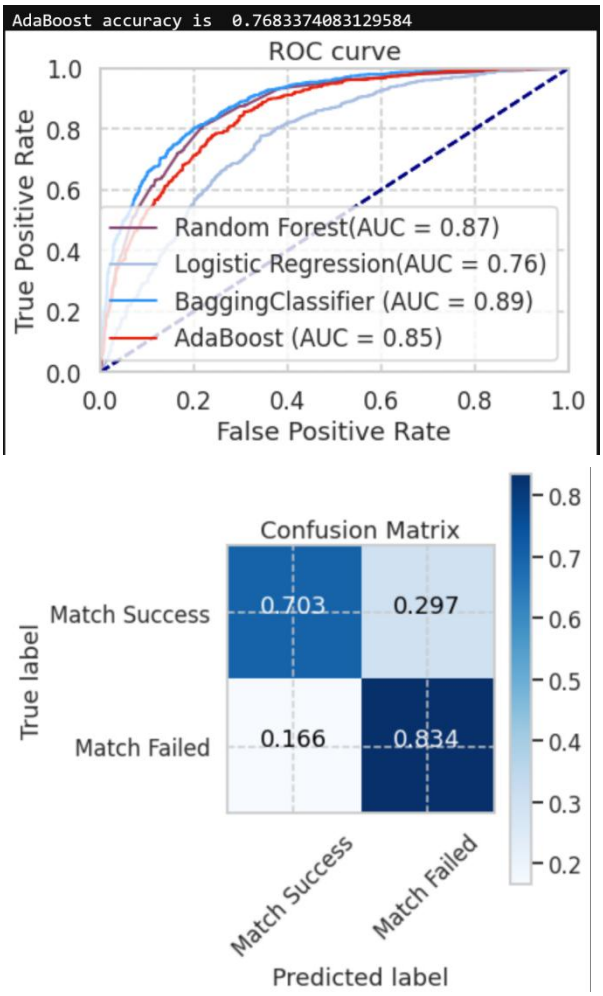


图 15　Adaboost 的 ROC 曲线和混淆矩阵

5. SVM

```
print(grid.best_params_)
print("---------------")
print(grid.best_estimator_)
```

```
{'C': 10}
---------------
SVC(C=10, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

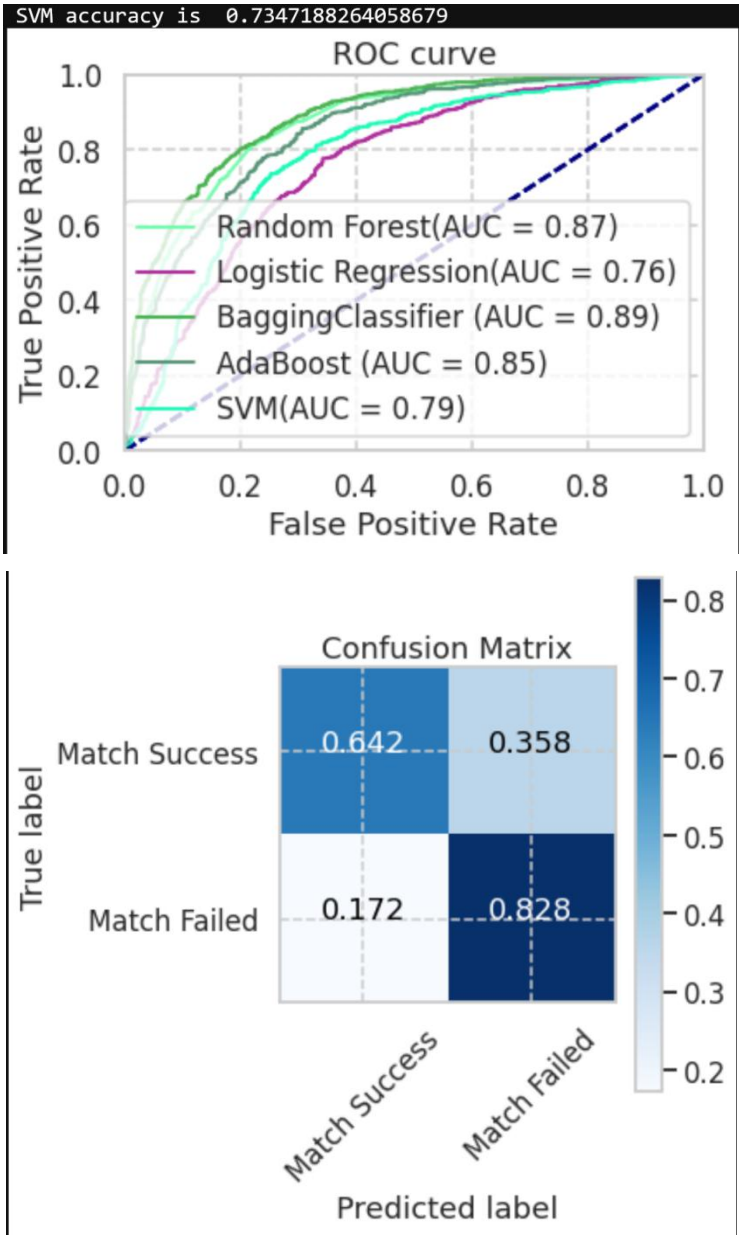


图 16　SVM 的 ROC 曲线和混淆矩阵

五种监督学习效果对比：

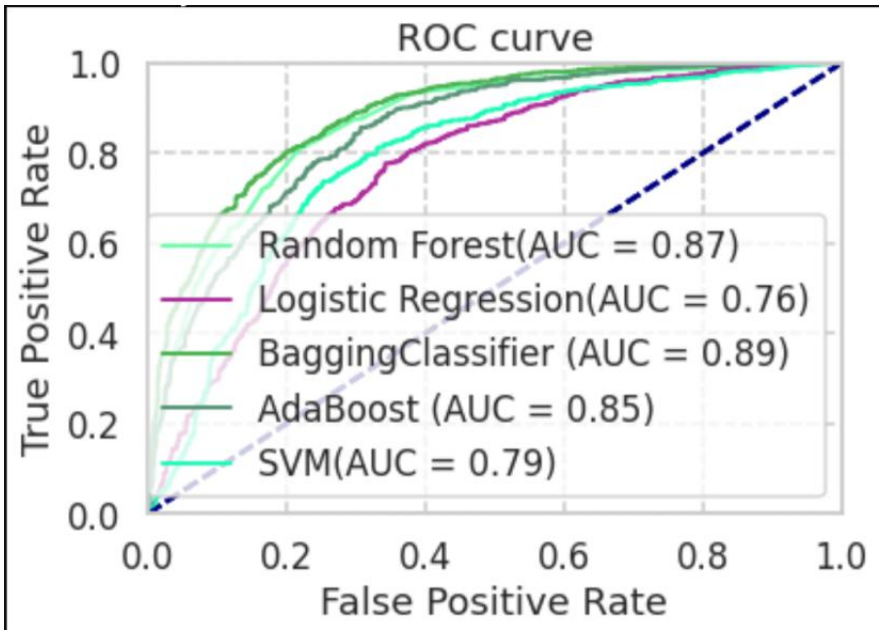


图 17 五种有监督模型效果对比

我们发现 BaggingClassifier 的效果最佳，随机森林和 Adaboost 稍逊但效果差不多好，而逻辑回归和支持向量机的效果都不佳。下面我们再对除支持向量机之外的四种模型做半监督学习-self-training 的对比。

（四）半监督学习

由于半监督学习每次都会生成许多新的带标签数据，因此在这里我们仅对每个有监督学习的模型单独做半监督学习的对比，且在每次对比前都从头开始重新处理数据、过采样、随机打乱，因此在这里也重新列出了每个有监督模型的 accuracy，值与前面（三）列出的不同。

这里半监督学习算法我们采用 self-training，即先用带标签数据训练出一个模型，用该模型对无标签数据做预测，取预测值中置信度较大的一部分与其对应的特征一起拿出来加入有标签数据中，在无标签数据中将这部分特征数据去掉，再次循环上述操作直至所有无标签数据都被去掉。

在这里置信度选择每个模型对应的 predict_proba()函数得到的值，将其降序排列后取其 index，对应到 y_predict_unlabeled 与 X_train_unlabeled 相应的值。

下面是随机森林模型半监督学习部分代码：

```python
#self-training
#考虑使用随机森林模型
from pandas import Series, DataFrame

semi_randomForest = RandomForestClassifier(
    bootstrap=True,
    criterion="gini",
    max_features=rand.best_estimator_.max_features,
    random_state=0)

for j in range(0,80):
    semi_randomForest.fit(X_train_labeled,y_train_labeled)
    y_predict_unlabeled = semi_randomForest.predict(X_train_unlabeled)
    #print(y_predict_unlabeled)#0-7999

    a = semi_randomForest.predict_proba(X_train_unlabeled)[:,1]
    sorted_nums = sorted(enumerate(a), key=lambda x: x[1],reverse=True)
    idx = [i[0] for i in sorted_nums] #把置信度按降序排列之后将对应的索引值存储在idx中

    n=len(y_train_labeled)
    for i in range(0,100):
        y_train_labeled[n+i]=y_predict_unlabeled[idx[i]]#有标签的第1000个y增加了
        df_insert = pd.DataFrame({'attr_o':[X_train_unlabeled.iloc[idx[i],0]],'sinc_o':[X_train_unlabeled.iloc[idx[i],1]],'intel_o':[X_train_unlabeled.i
                                  ,'fun_o':[X_train_unlabeled.iloc[idx[i],3]],'amb_o':[X_train_unlabeled.iloc[idx[i],4]],'shar_o':[X_train_unlabeled.ilo
        #iloc 用的是相对值(idx从0-7999,无标签从2000-9999)
        X_train_labeled = X_train_labeled.append(df_insert,ignore_index = True) #有标签的第1000个X增加了
        X_train_unlabeled.drop(2000+idx[i])
    X_train_unlabeled.reset_index(drop=True)
print(len(X_train_labeled))
print(len(y_train_labeled))
```


图 18 随机森林半监督学习代码

1. 随机森林

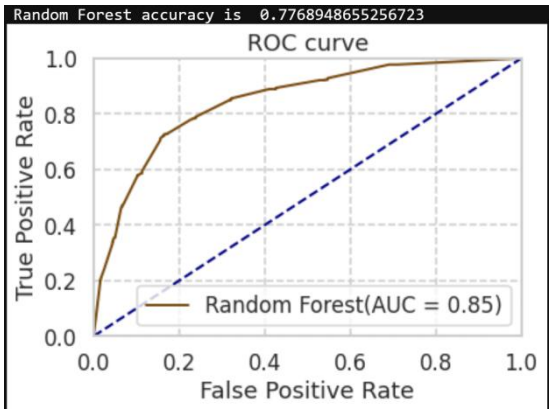


图 19 随机森林有监督学习 Accuracy 与 ROC 曲线

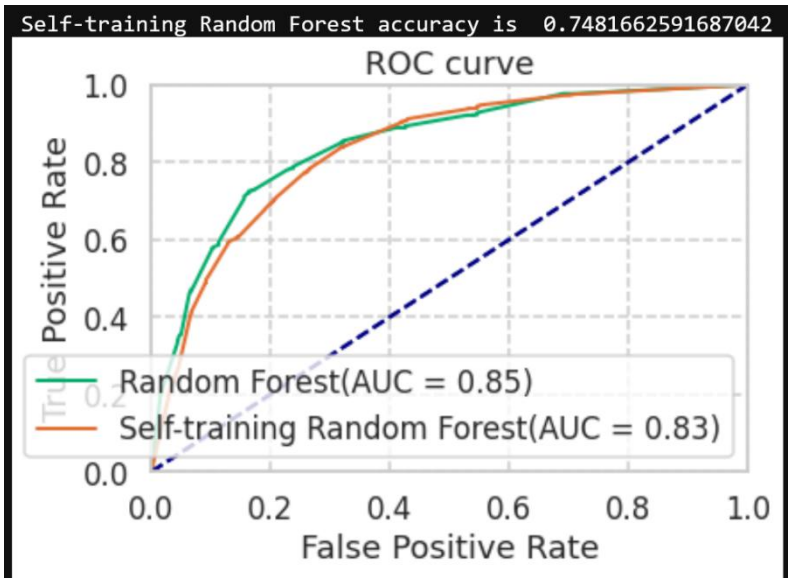


图 20 随机森林半监督学习 Accuracy 与 ROC 曲线

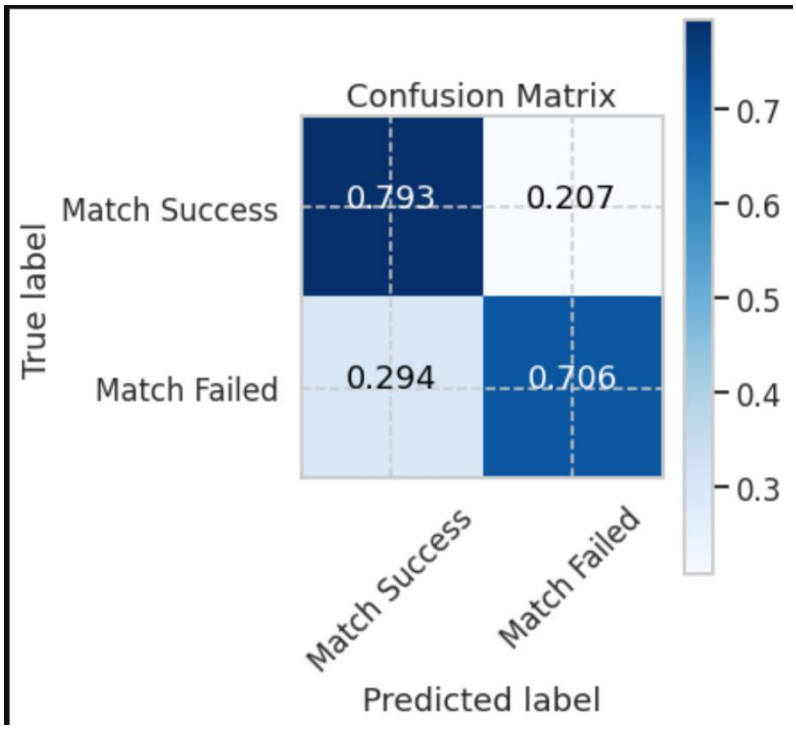


图 21 随机森林半监督学习混淆矩阵

我们发现对于预测效果较好的随机森林模型，应用半监督学习后准确性反而变差了。

训练时间对比：

用 2000 条带标签数据做有监督学习：

```
train_rf_time :  0.03609609603881836
```

用 2000 条带标签数据、8000 条无标签数据做半监督学习：

```
print("train_semi_rf_time : ",train_semi_rf_time)
print(len(X_train_labeled))
print(len(y_train_labeled))
```

```
train_semi_rf_time :  27.7919020652771
10000
10000
```

我们发现训练时间长了 9000 倍，但是效果下降了。

2. 逻辑回归

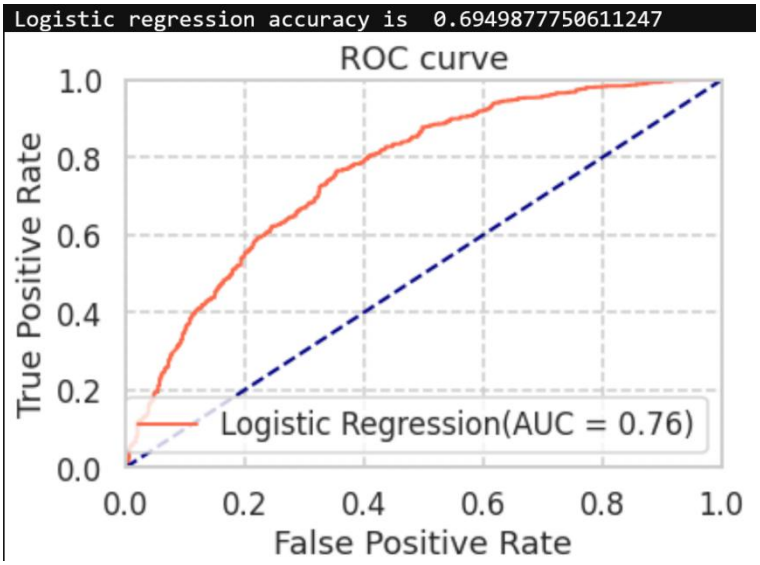


图 22 逻辑回归有监督学习 Accuracy 与 ROC 曲线

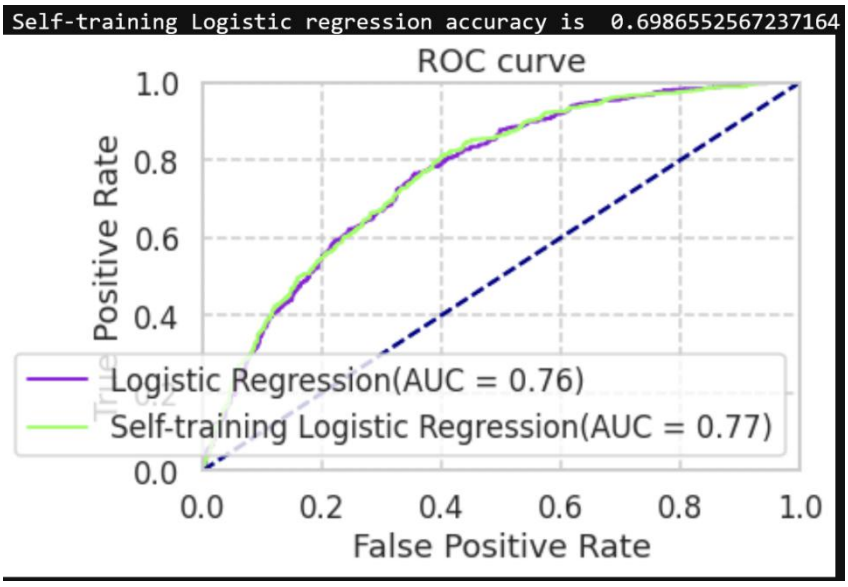


图 23 逻辑回归半监督学习 Accuracy 与 ROC 曲线

发现对于原本效果很差的逻辑回归，半监督学习的准确性提高了约 0.003，但时间成本仍增加了许多。

```
train_lr_time :  0.0057909488677978516
train_semi_lr_time :  37.279953479766846
```

3. Bagging Classifier

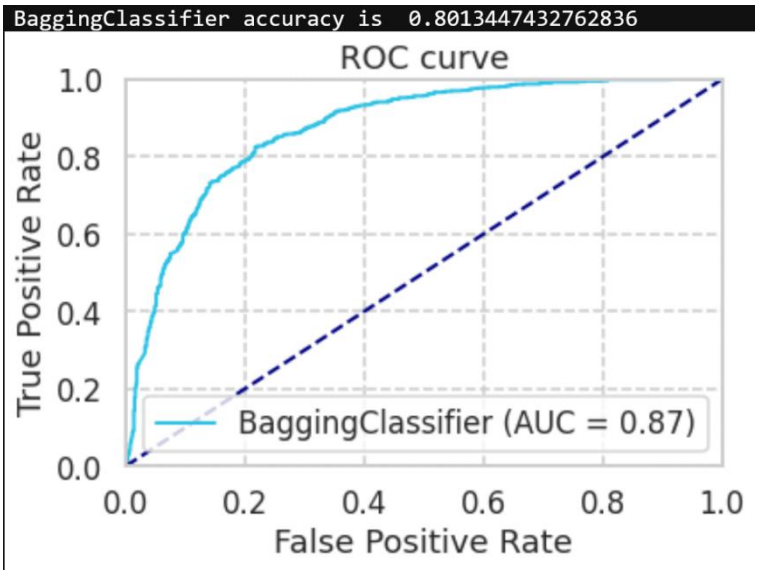


图 24 Baggingclassifier 有监督学习 Accuracy 与 ROC 曲线

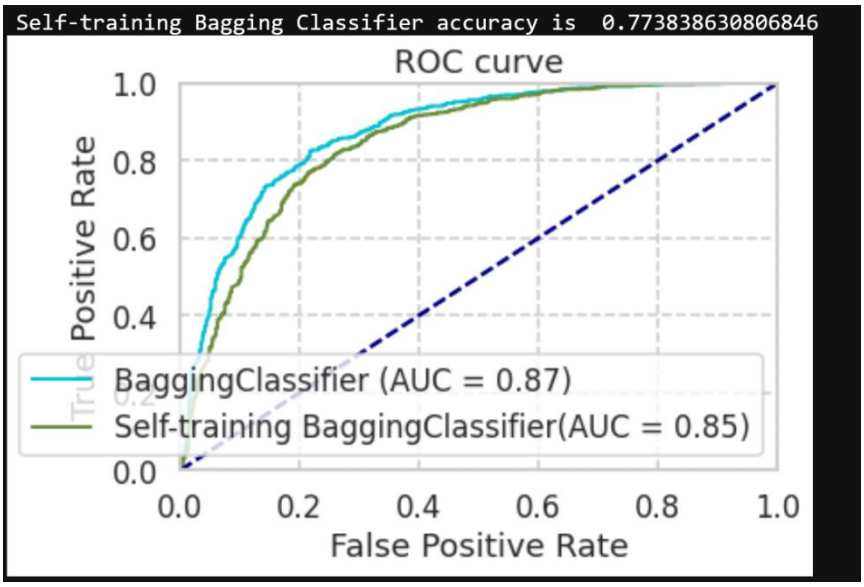


图 25Baggingclassifier 半监督学习 Accuracy 与 ROC 曲线

我们发现对于原来分类效果较好的 Baggingclassifier 而言，半监督学习的效果较差。

4. Adaboost

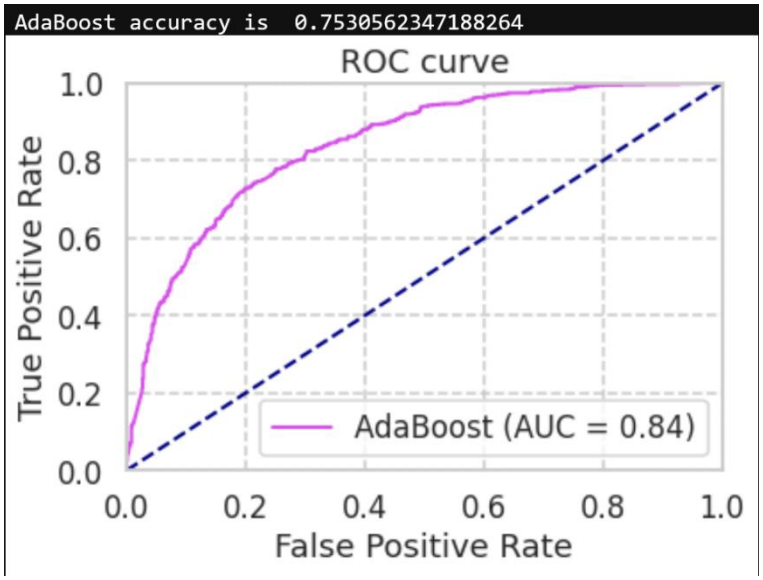


图 26 Adaboost 有监督学习 Accuracy 与 ROC 曲线

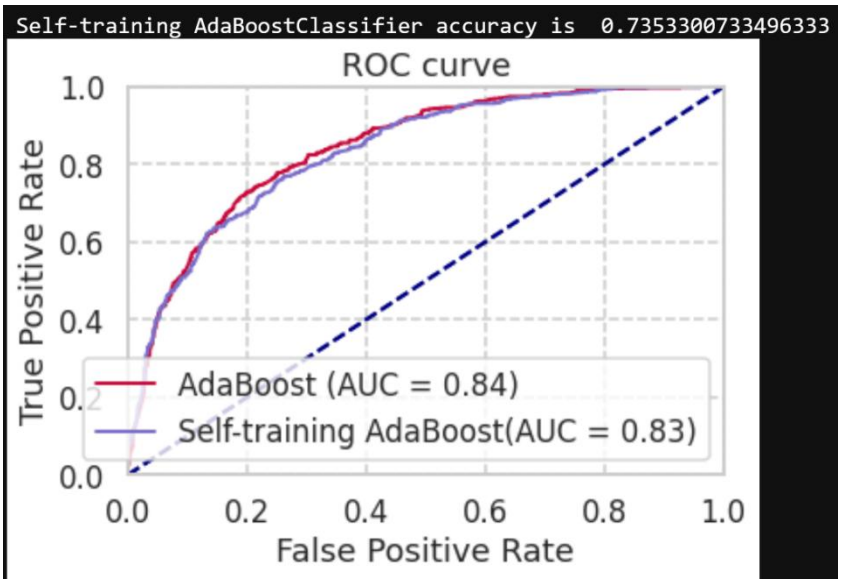


图 27 Adaboost 半监督学习 Accuracy 与 ROC 曲线

我们发现对于原来分类效果较好的 Adaboost 而言，半监督学习的效果较差。

四、分析与讨论

1. 我们发现对于大多数上述模型而言，半监督学习不能提高分类效果或提高效果很不明显。

对此我认为，当面对大量的数据时，半监督学习方式通常不能实现和监督学习中所实现的相同渐近性质，未标注的数据反而会引入偏差。这些非监督方法学到的特征可能并不是分类器真正需要的特征。

有监督模型在无标签数据上得到的预测值泛化性能不好，尤其是当带标签数据的分布不是严格一致而是像本次项目中有许多人为因素、扰动、噪声时，预测得到的结果就与带标签数据不具有很相似的分布，造成最后预测效果变差。

2. 而回到任务本身，我们确实发现吸引力、真诚、聪明才智、有趣、有目标雄心，有同样兴趣爱好这些特征在相亲匹配成功上发挥了作用，这对婚恋匹配的研究有所帮助。

3. 同时在实验过程中发现，如果我们不进行过采样，在实验数据如此不均衡的条件下会导致 FP 远大于 TN，FP 值与 TP 值近乎相同，拟合效果很差。我们发现了过采样、欠采样技术的重要性。

五、总结与寄语

在本次半监督学习与有监督学习的对比中了解到了半监督学习算法在实际不同数据场景下的局限性，希望在今后的学习研究中，可以研究发现其他新的泛化能力更好的半监督学习算法来真正有效的解决带标签数据过少的问题，提高模型分类效果。同时，发现吸引力、真诚、聪明才智、有趣、有目标雄心，有同样兴趣爱好对与相亲匹配有重要影响，对以后婚恋有指导作用。