机器学习纳米学位

毕业项目:Human or Robot?

夏强

2017年2月7日

I. 问题的定义

(大概 1-2 页)

1.1 项目概述

项目将根据拍卖数据识别其中的机器人用户，这是Facebook在2015年发起的工程师比赛，比赛的前几名有机会参加Facebook的面试。

两个项目数据集：

1. 不同拍卖的七百六十万次出价情况包括出价id，出价人id，拍卖id，货物类别，使用设备，出价时间，ip地址，ip归属国，url。*其中敏感信息均被模糊化处理*

2. 及出价人数据包括id，支付账号，地址，这个账号是否是机器人（训练集包含此项，测试集不包含）。*其中敏感信息均被模糊化处理*

项目目标： 从拍卖者中识别出机器人，方便网站标记删除防止拍卖中的不公平交易的发生。

1.2 问题陈述

问题大致分为三部分：

1. 特征提取：从拍卖出价数据中提取出用户特征。例如通过清洗出价信息清晰出每个拍卖者的参与拍卖数，出价数，每场出价数，出价在不同的登陆国家，ip，url，登陆设备中的分布等特征。
2. 特征工程：根据数据特征分布规律，缺失值，类别是否均衡对数据进行处理。例如对缺失值进行填充，数据变换使分布更接近正太分布，通过特征选择或者降维以减少无关信息。
3. 模型选择与优化：根据模型的性能选择合适的模型对数据进行预测

1.3 评价指标

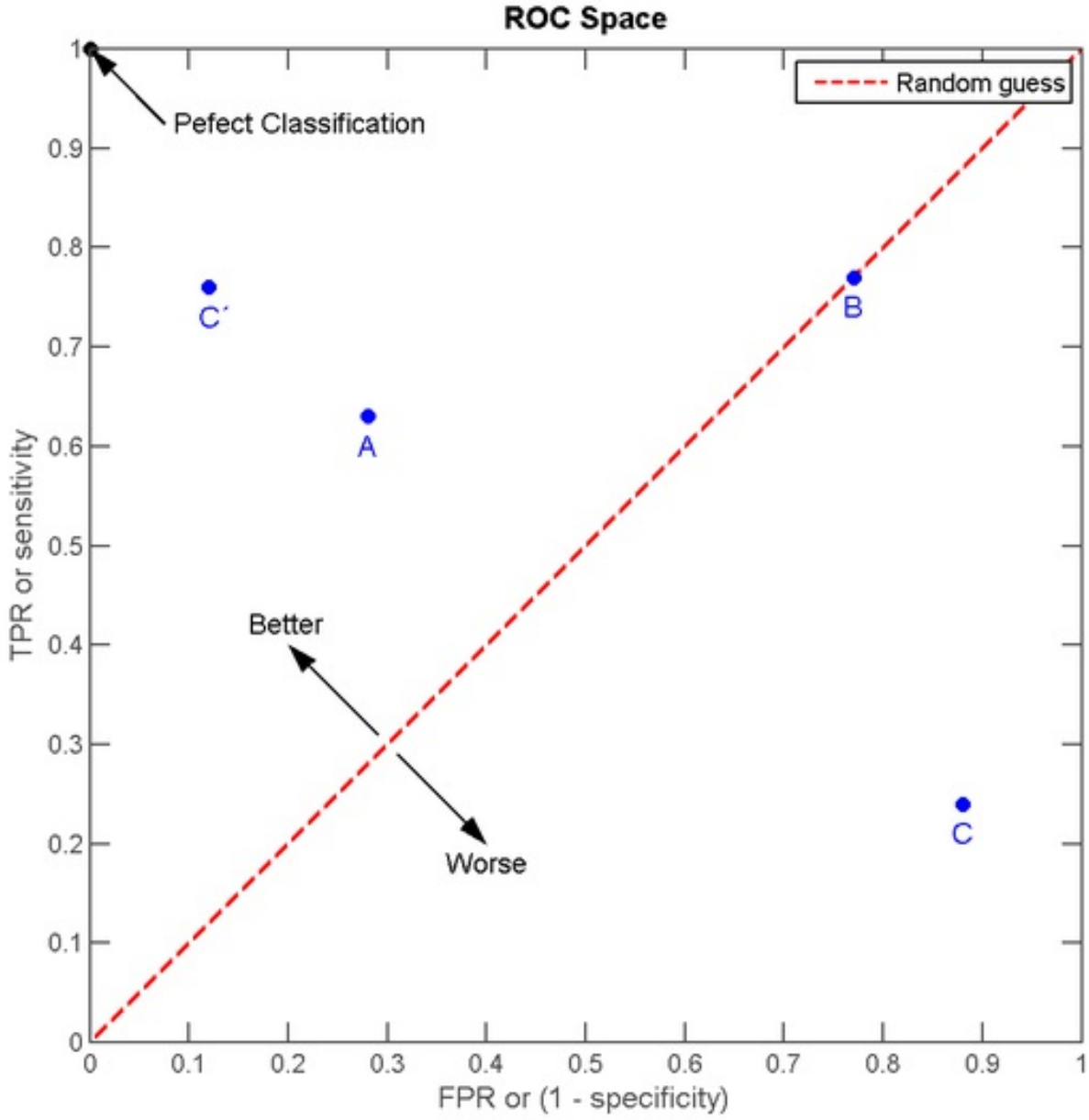
本文中使用roc曲线下的面积作为评价指标

一个二分类问题,输出结果只有两种类别的模型,例如：（阳性／阴性）（有病／没病）（垃圾邮件／非垃圾邮件）（0／1）

对于这个问题的预测就有四种情况（以检测是否高血压为例）：

1. 真阳性（TP）：诊断为有，实际上也有高血压。
2. 伪阳性（FP）：诊断为有，实际却没有高血压。
3. 真阴性（TN）：诊断为没有，实际上也没有高血压。
4. 伪阴性（FN）：诊断为没有，实际却有高血压。

ROC空间 将伪阳性率（FPR）定义为 X 轴，真阳性率（TPR）定义为 Y 轴 率形成的空间



完美的预测：（1,0）点X=0 代表着没有伪阳性，Y=1 代表着没有伪阴性

随机的预测：从 (0, 0) 到 (1, 1) 对角线，即随机猜测线

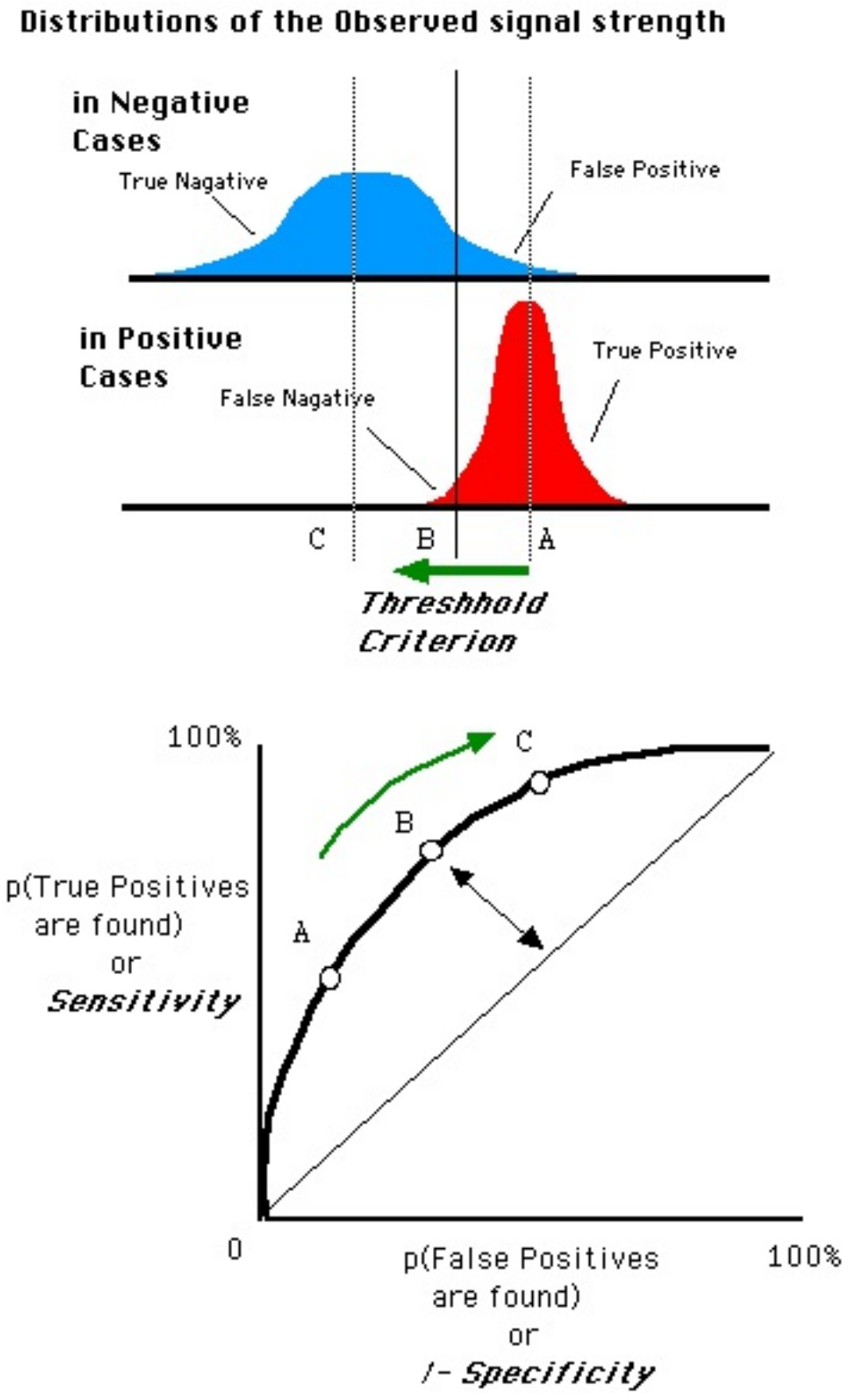
在roc空间中点与随机猜测线的距离是衡量预测能力的指标，离左上角的点（1,0）越近预测越准确。

ROC曲线 ROC空间的单个点是在给定阈值下的(FPR, TPR)坐标，所有阈值下的点就形成了ROC曲线

假设采用逻辑回归分类器，其给出针对每个实例为正类的概率，那么通过

设定一个阈值如0.6，概率大于等于0.6的为正类，小于0.6的为负类。对应的就可以算出一组(FPR,TPR),在平面中得到对应坐标点。随着阈值的逐渐减小，越来越多的实例被划分为正类，但是这些正类中同样也掺杂着真正的负实例，即TPR和FPR会同时增大。阈值最大时，对应坐标点为(0,0),阈值最小时，对应坐标点(1,1)。

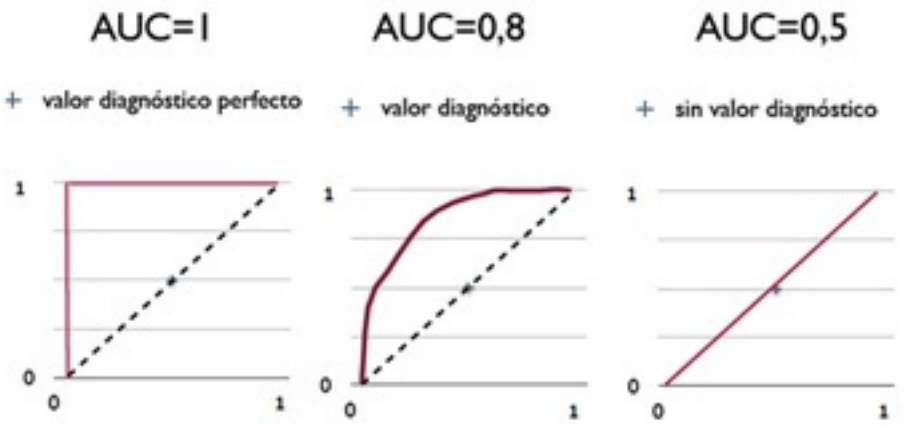
如下面这幅图，为ROC曲线，线上每个点对应一个阈值。



ROC曲线下方的面积（Area under the Curve of ROC (AUC)）

从AUC判断分类器（预测模型）优劣的标准:

1. AUC = 1，是完美分类器，采用这个预测模型时，存在至少一个阈值能得出完美预测。绝大多数预测的场合，不存在完美分类器。
2. 0.5 < AUC < 1，优于随机猜测。这个分类器（模型）妥善设定阈值的话，能有预测价值。
3. AUC = 0.5，跟随机猜测一样（例：丢铜板），模型没有预测价值。
4. AUC < 0.5，比随机猜测还差；但只要总是反预测而行，就优于随机猜测。



*AUC计算*常用的方法有梯形法，ROC AUCH法

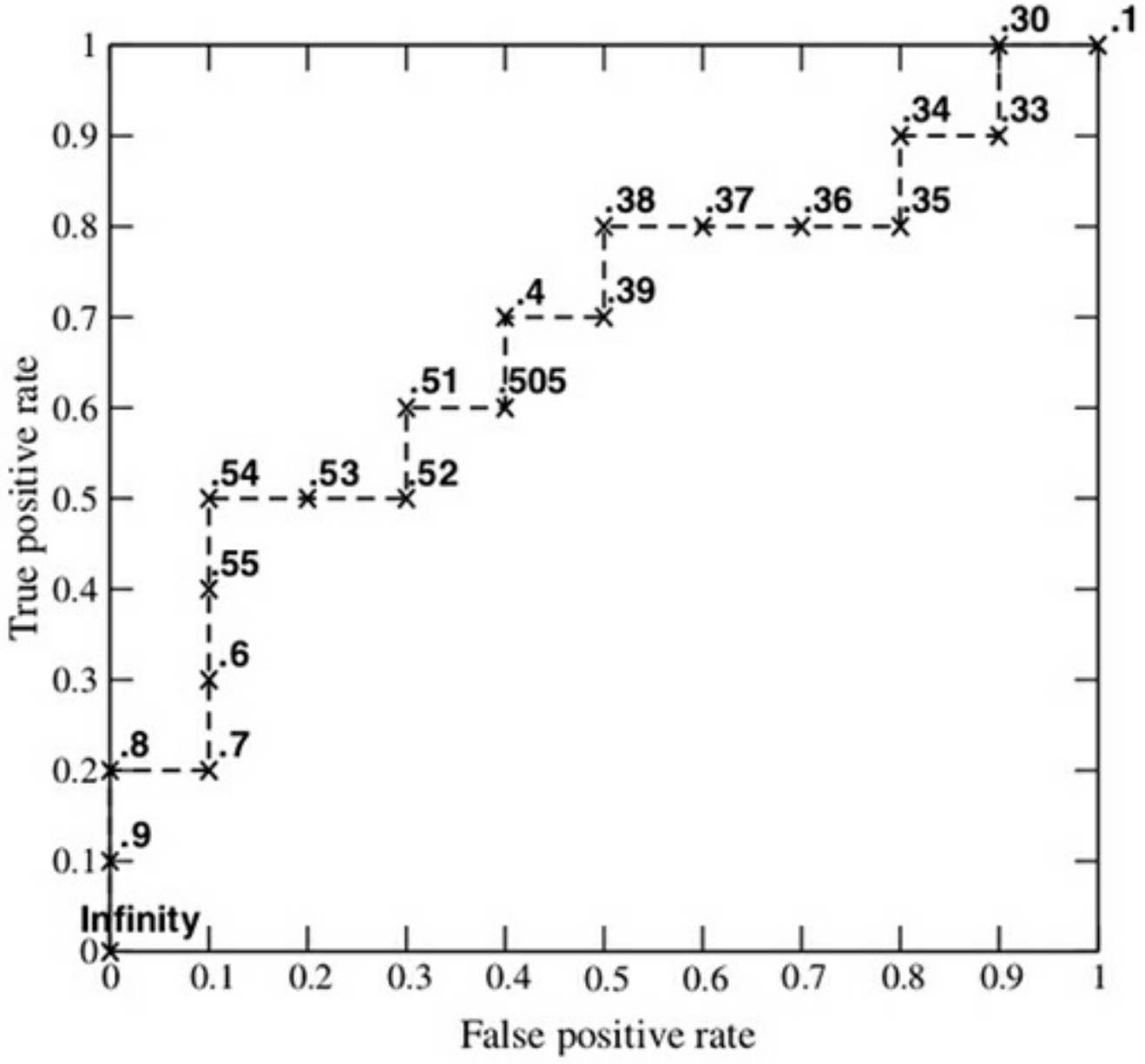
梯形法（英语：trapezoid method）：简单地将每个相邻的点以直线连接，计算连线下方的总面积。

示例

下图是一个示例，图中有20个样本，“Class”表示每个样本真正的标签（p表示正样本，n表示负样本），“Score”表示每个测试样本属于正样本的概率

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

接下来，我们从高到低，依次将“Score”值作为阈值threshold，当测试样本属于正样本的概率大于或等于这个threshold时，我们认为它为正样本，否则为负样本。每次选取一个不同的threshold，我们就可以得到一组FPR和TPR，即ROC曲线上的一点。这样一来，我们一共得到了20组FPR和TPR的值，将它们画在ROC曲线的结果如下图：



然后通过梯形法即可计算AUC

II. 分析

(大概 2-4 页)

2.1 数据的探索及可视化

2.1.1 探寻数据中的问题

缺失值

```
bot_or_human.isnull().sum().sort_values(ascending=False)
```

```
bids_count_lasthalfh        3788
dt_std2                     1677
dt_std1                     1178
dt_mean2                    1156
dt_min2                     1156
country_entropy              104
dt_mean1                     103
dt_min1                      103
bids_count                    99
auctions_count                99
bids_per_auction_median       99
ip_auction_max                99
countries_auction_median      99
countries_auction_mean        99
countries_auction_max         99
ips_count                     99
ip_entropy                    99
```

与出价相关次数的特征均有99个缺失值说明，有99个用户没有进行出价，发现有99人没有出价，故其出价数，ip，url数等特征实际上为0故用0填充。 最后半小时出价数的缺失值也是由于实际上是他在最后半小时没有出价造成的也是0填充。 其他缺失值则使用均值填充。

数据分布倾斜

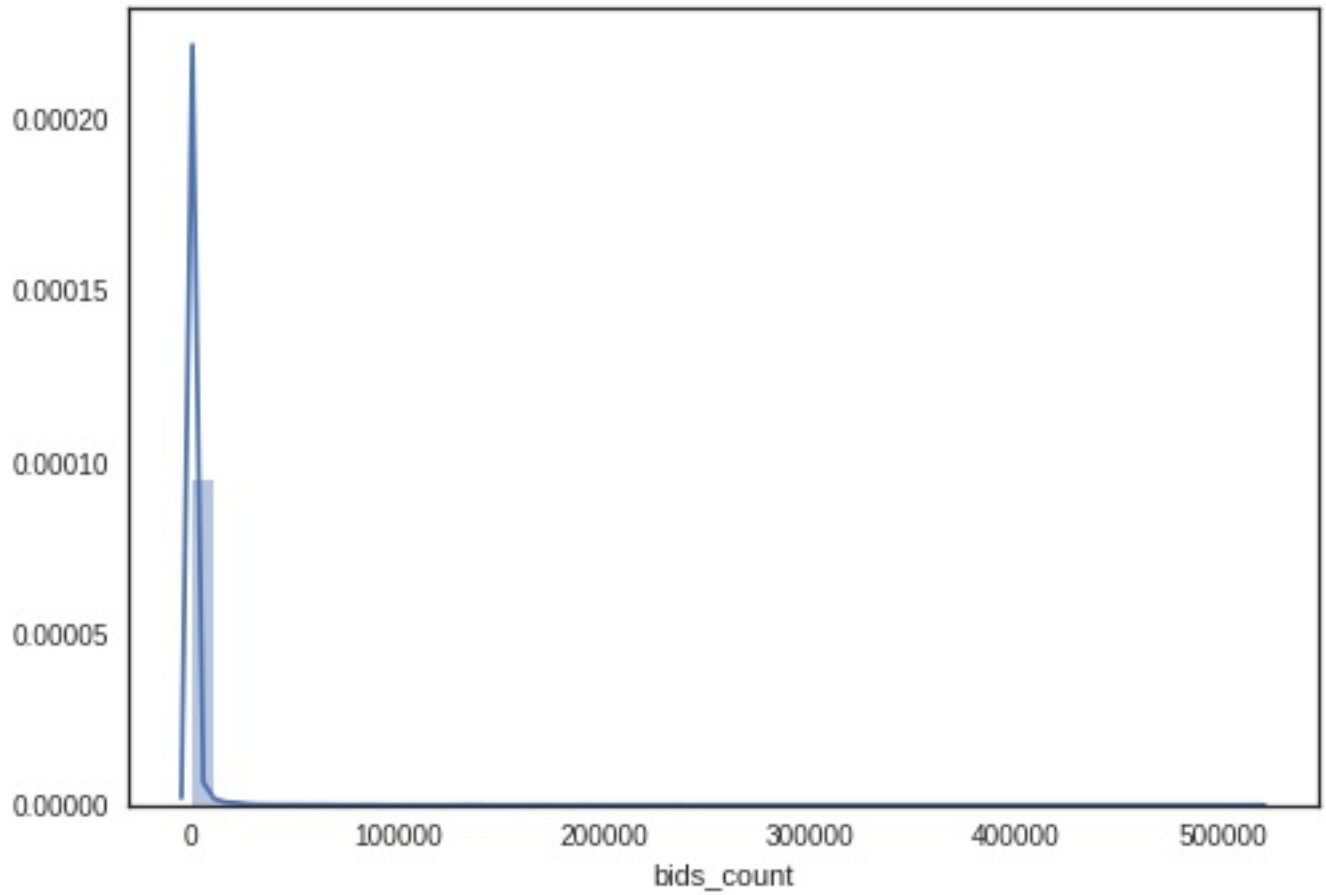
数据集中总共有4700个测试数据（未标注是否是机器人）和2013个训练数据，前面经过清洗得到35个特征。

描述性统计的发现： **一些特征的数据之间大小相差极大** 如图

```
# 描述性统计：
bot_or_human.describe()
```

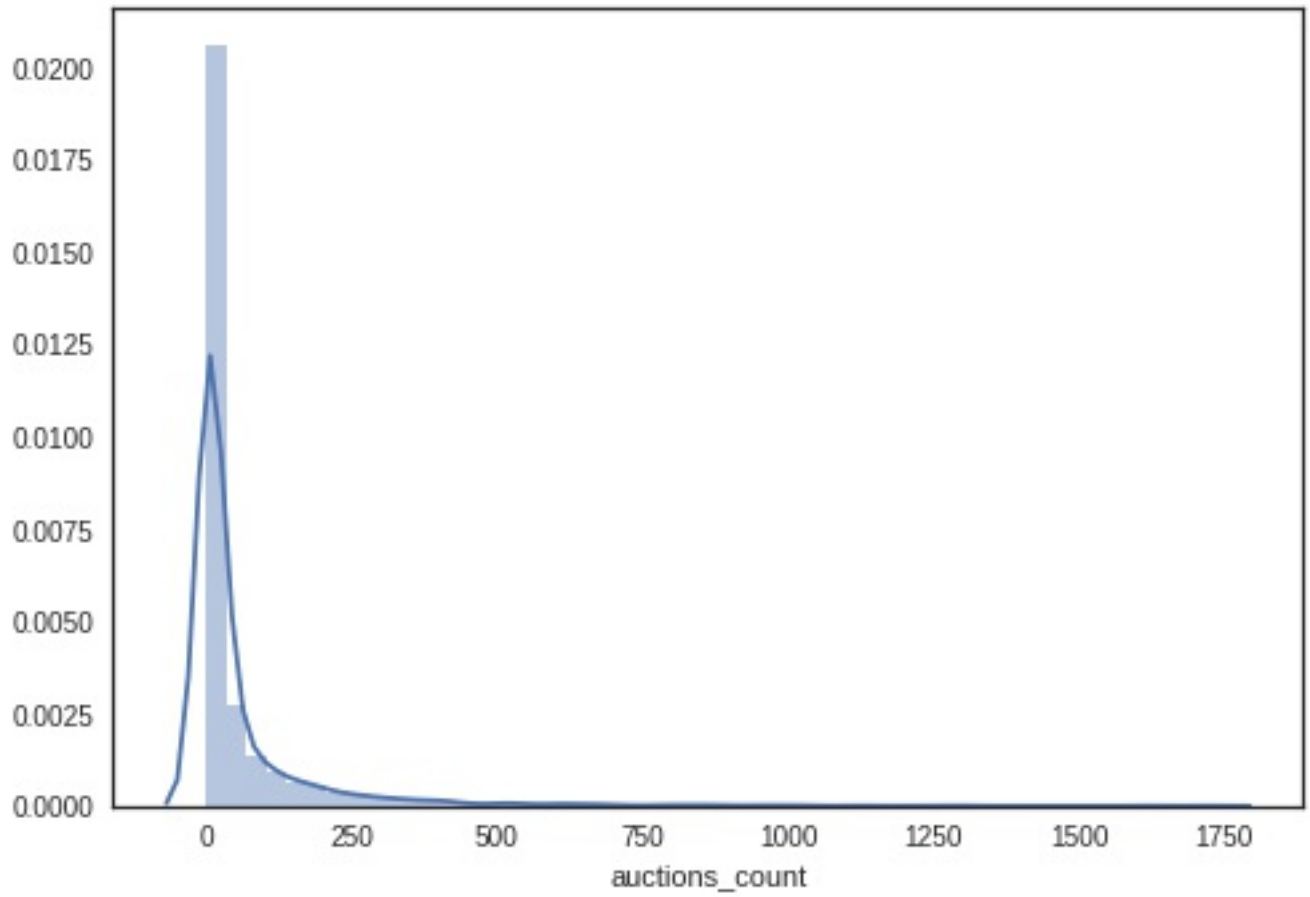
	outcome	payment_address_same	bids_count	auctions_count	bids_per_auction_median	countries_count
count	6713.000000	6713.000000	6713.000000	6713.000000	6713.000000	6713.000000
mean	-0.684791	0.140921	1140.523462	56.955311	2.708476	12.536571
std	0.496564	0.347966	9523.031905	130.983566	20.467453	22.436327
min	-1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	-1.000000	0.000000	3.000000	2.000000	1.000000	1.000000
50%	-1.000000	0.000000	17.000000	9.000000	1.000000	3.000000
75%	0.000000	0.000000	178.000000	46.000000	2.000000	12.000000
max	1.000000	1.000000	515033.000000	1726.000000	1125.000000	178.000000

比如用户出价数，均值为1140,中位数为17,最大值有515033。



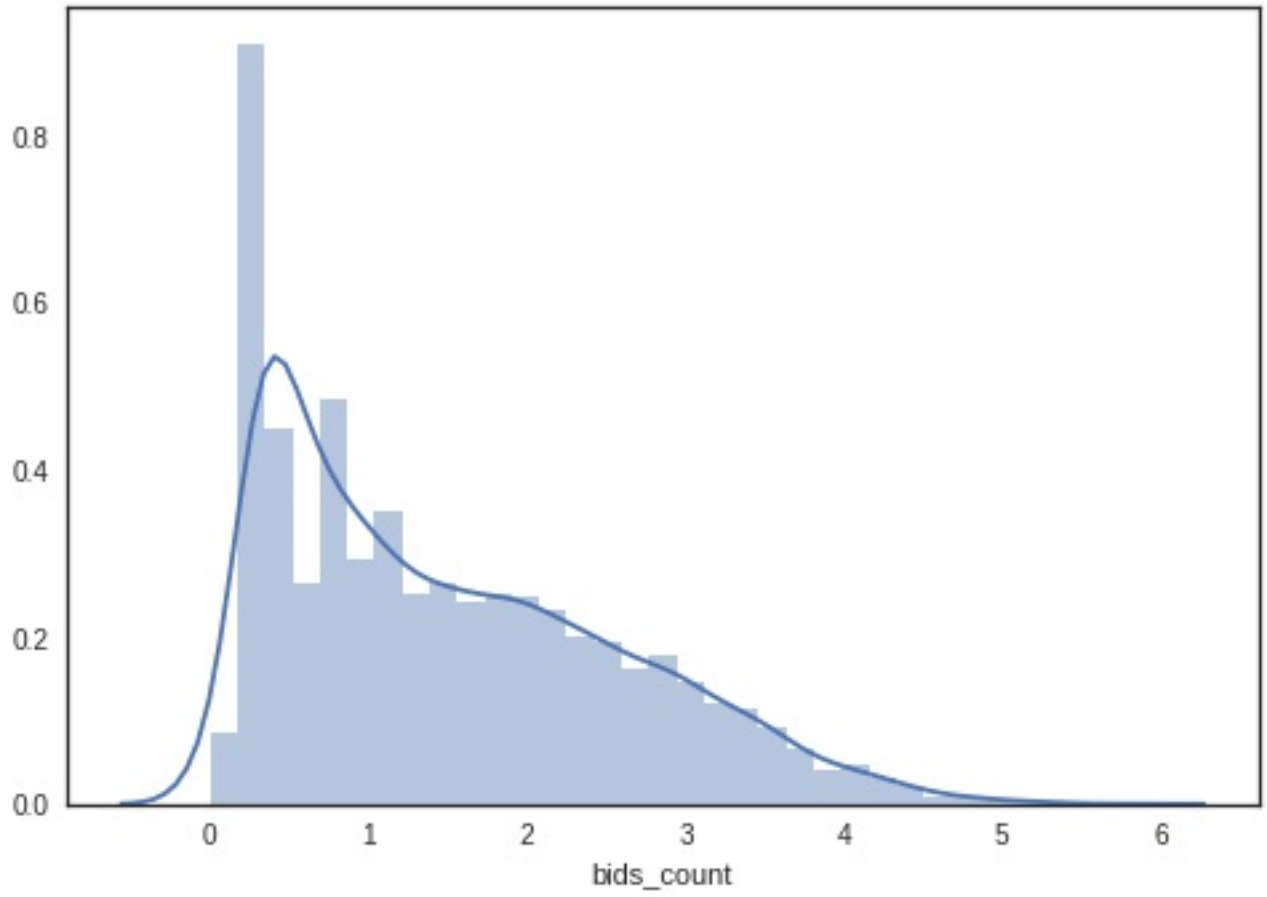
出价数分布图

用户参与拍卖数平均值为56.955中位数为9，最大值1726

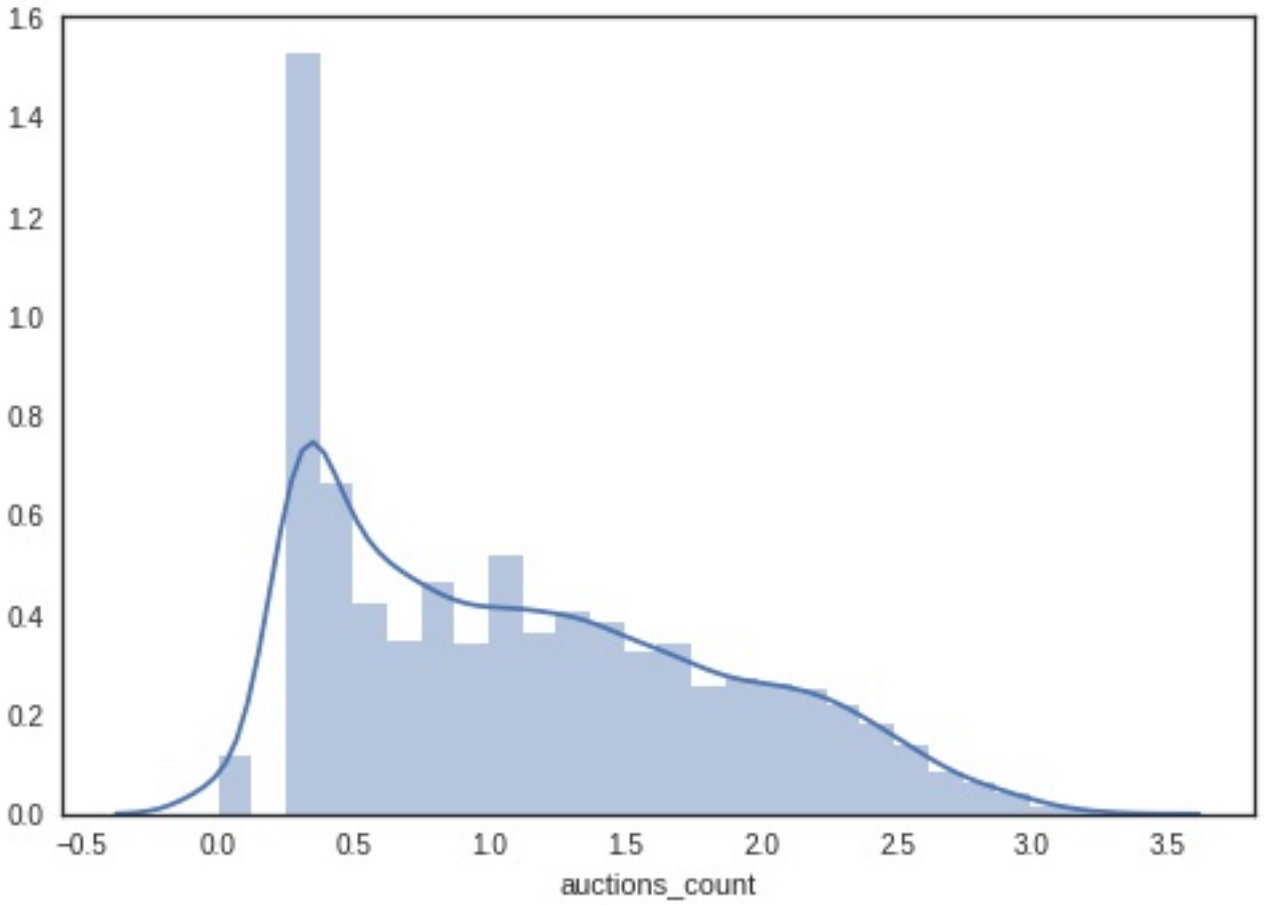


参与拍卖次数分布图

对数据进行np.log10(x+1)运算后分布图如下



np.log10(x+1)后的出价次数

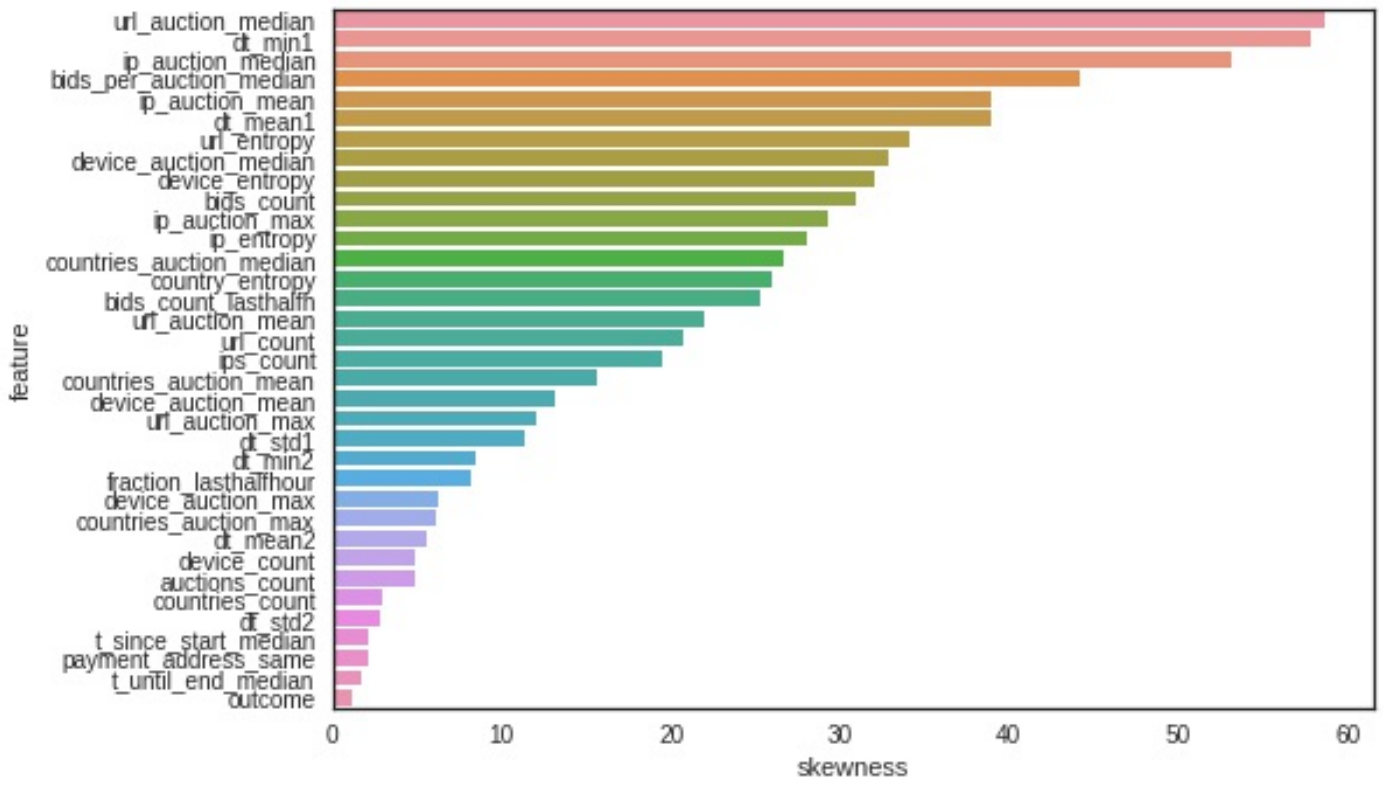


np.log10(x+1)后的参与拍卖次数

偏态（skewness） 是数据对称性的度量，当偏态系数大于 1 或者小于-1 是，数据的分布可以称之为高度偏态分布． 数据各个特征的偏态如下

```
In [8]: bot_or_human.skew()
```

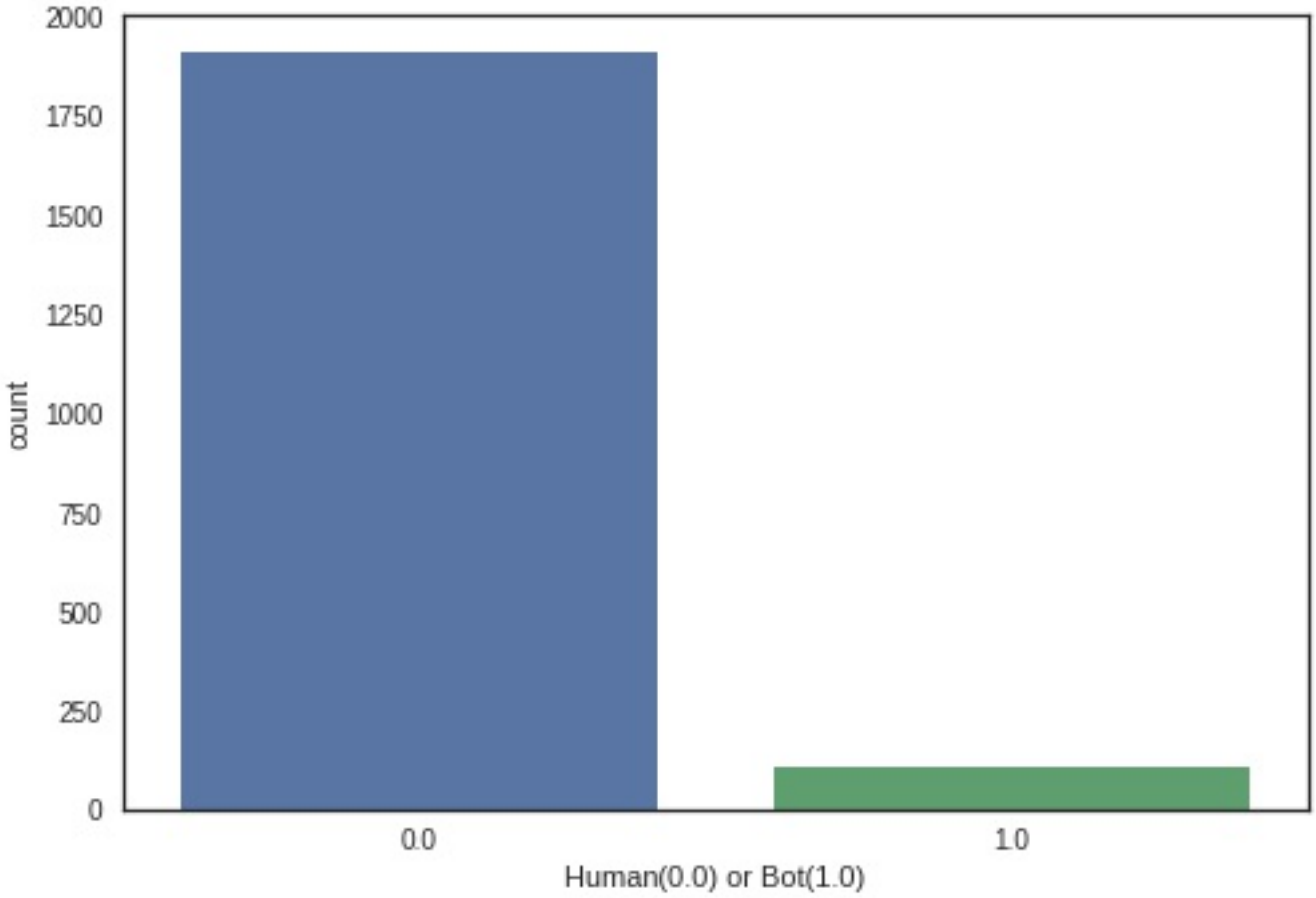
```
Out[8]: outcome                      1.166939
        payment_address_same         2.064494
        bids_count                  30.965704
        auctions_count               4.893641
        bids_per_auction_median     44.231919
        countries_count              3.005297
        country_entropy             26.004329
        countries_auction_median    26.684192
        countries_auction_mean      15.610466
        countries_auction_max        6.041154
        ips_count                   19.552029
        ip_entropy                  27.986003
        ip_auction_median           53.143528
        ip_auction_mean             38.914919
        ip_auction_max              29.288980
        url_count                   20.765973
        url_entropy                 34.064595
        url_auction_median          58.628736
        url_auction_mean            21.992065
        url_auction_max             12.055184
        device_count                 4.903581
        device_entropy              31.980816
        device_auction_median       32.795614
        device_auction_mean         13.093981
        device_auction_max           6.273373
        bids_count_1asthalfh        25.341878
        fraction_1asthalfhour        8.216133
        t_until_end_median           1.634776
        t_since_start_median         2.065964
        dt_min1                     57.844089
        dt_std1                     11.387861
        dt_mean1                    38.912956
        dt_min2                      8.461613
        dt_std2                      2.853804
```



所有特征的偏态系数均大于１，数据的所有特征分布均向右倾斜

解决方法：对数据取对数运算

类别不均衡问题



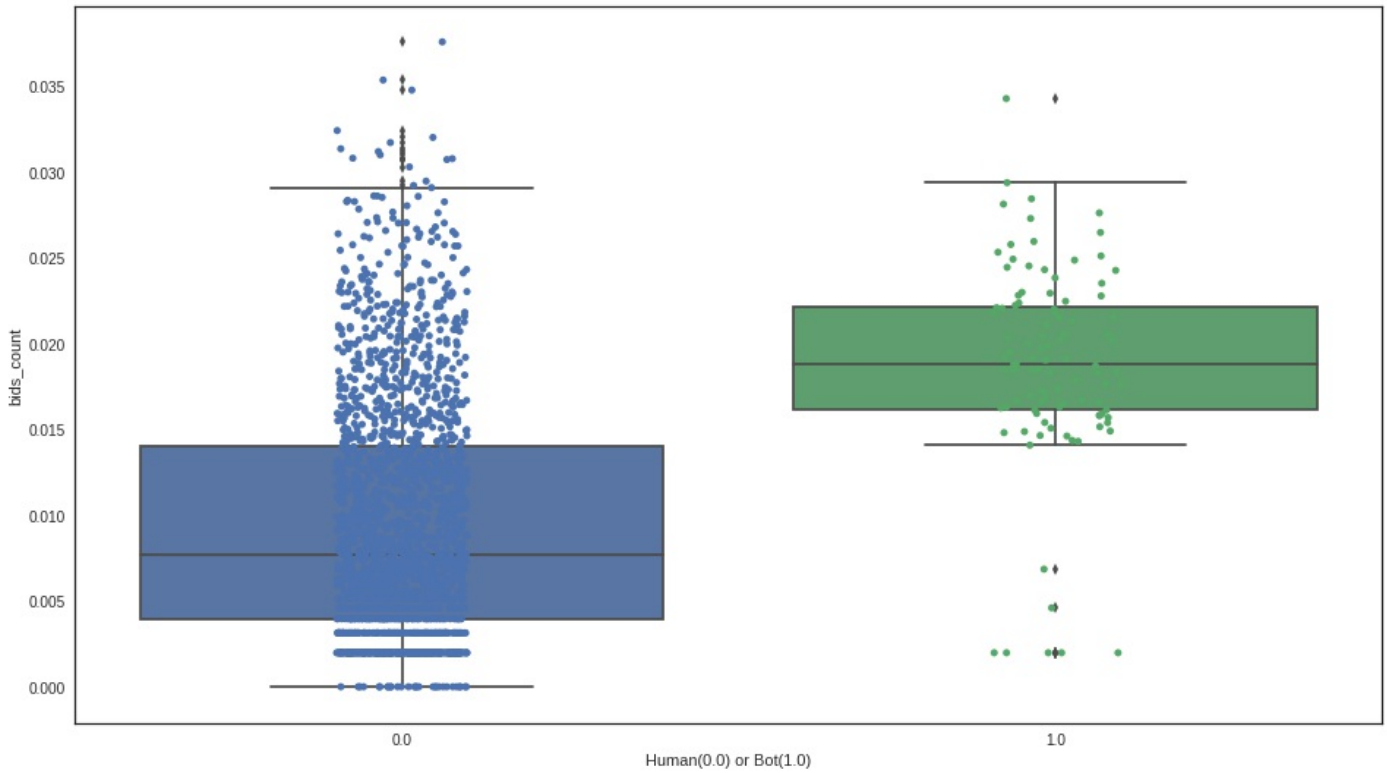
如图总共2013个训练数据有1910个人类和103个机器人，考虑到训练数据量较小将采用过采样的方法对数据进行处理·

2.1.2 双变量分析

- 注：以下分析均在修复缺失及偏态问题并归一化之后

出价次数

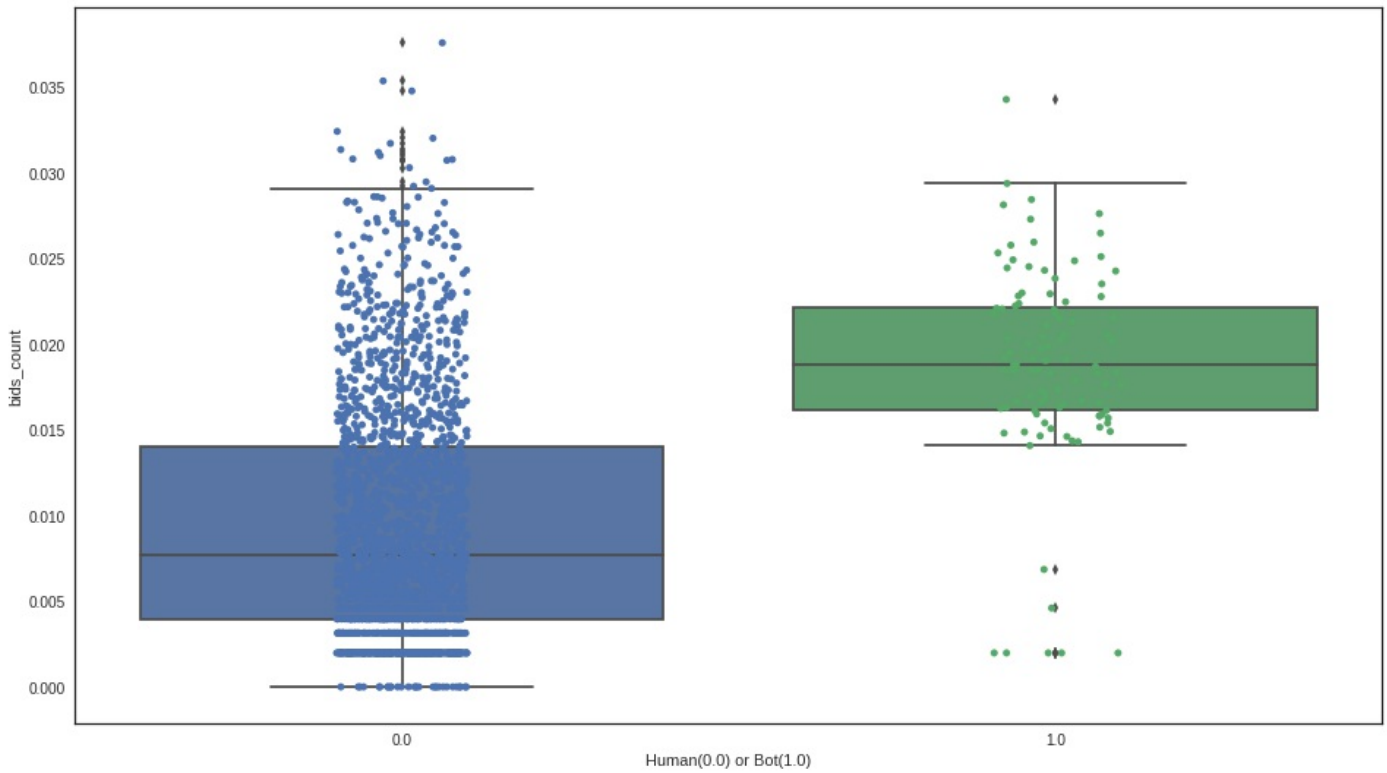
人类和机器人出价次数如下图



由图可知机器人出价次数相对与人类通常要高，其中位数上下四分位数均高于人类

出价次数

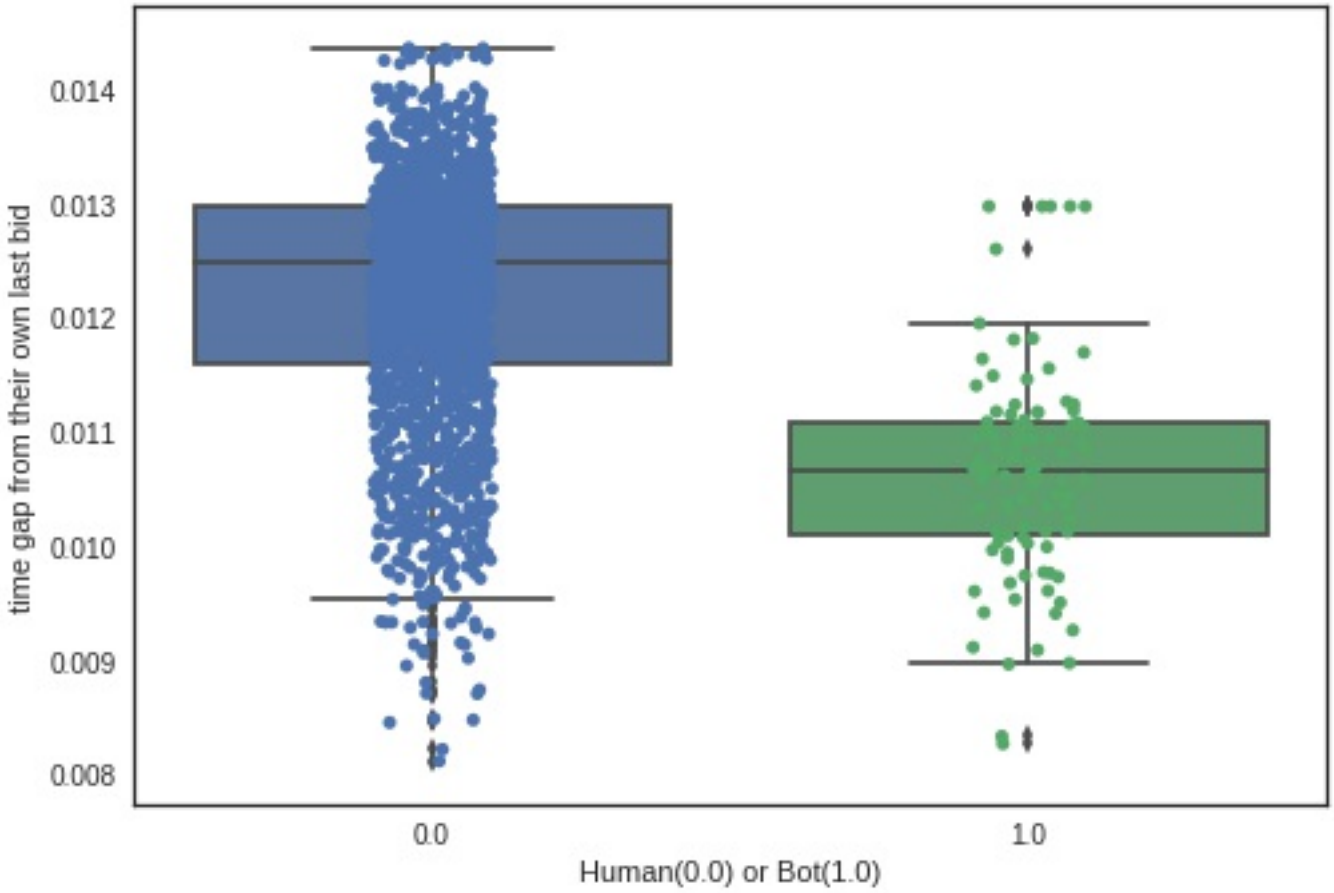
人类和机器人出价次数如下图



由图可知机器人出价次数相对与人类通常要高，其中位数上下四分位数均高于人类

与自己的出价间隔

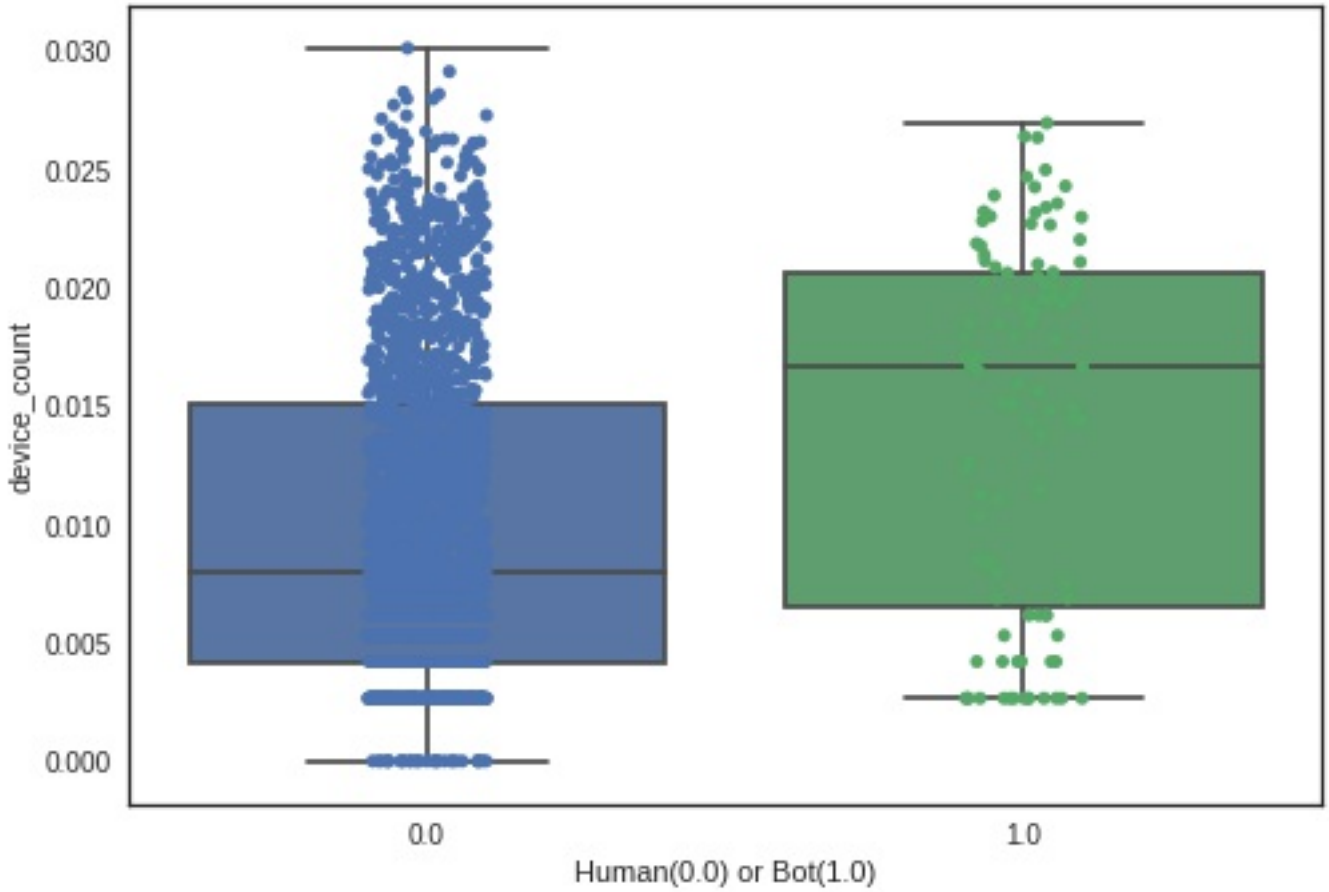
人类和机器人出价间隔



由图可知机器人比人类出价时间间隔更短

拍卖使用移动设备数

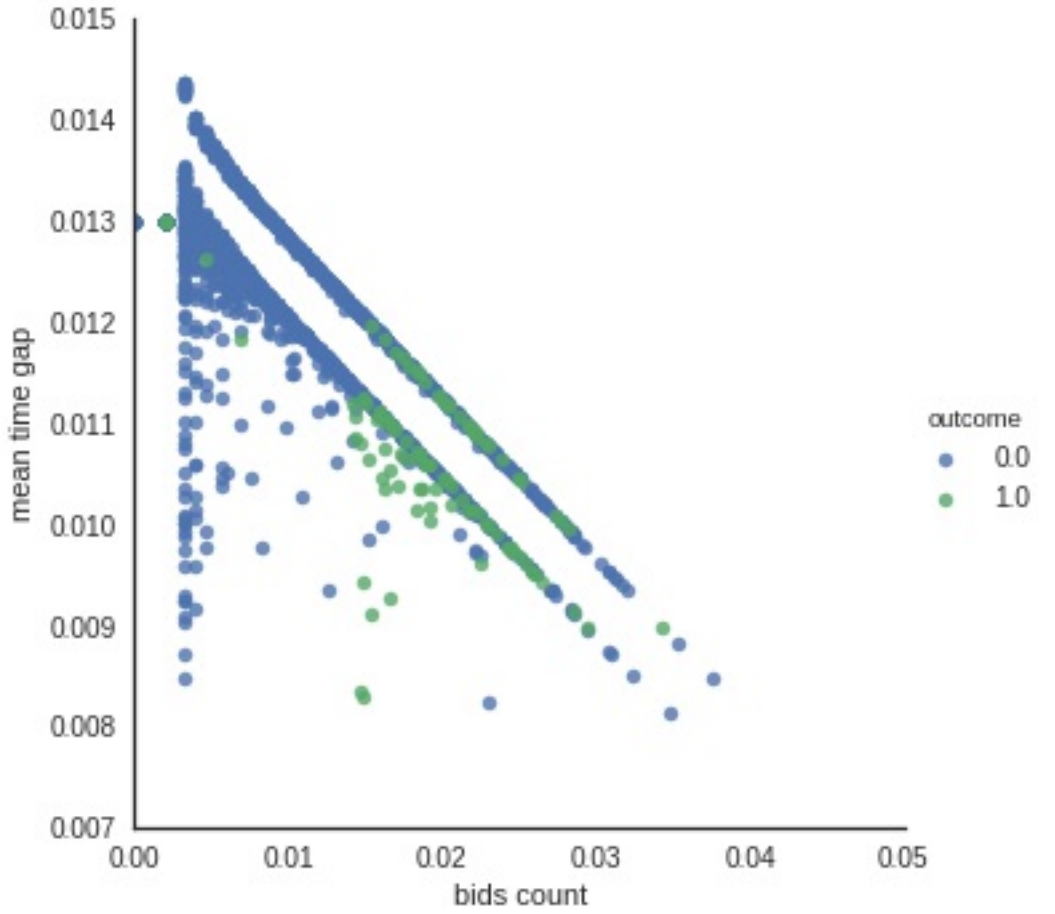
人类和机器人设备数



由图可知机器人通常比人类使用更多的移动设备来参加拍卖

2.1.3 多变量分析

出价间隔VS出价次数VS是否机器人



如图所示，机器人一般出现在出价多出价间隔短的竞拍着当中

2.2 算法和技术

应用到的机器学习算法有 随机森林random forest，极端随机树extremely randomized trees,梯度提升算法。

2.2.1 随机森林

决策树

决策树是一种对实例进行分类的树型结构。决策树通常是递归的选择最优特征对训练数据进行分割，使各个子数据集有一个最好的分类的过程。 在实际应用中以上方法生成的决策树常常会长的很深而对训练数据学习的太好却不能预测新数据，也就是过拟合。

Bagging

Bagging,在数据集中随机采样m次形成大小为m的采样集，采样出T个采样集，然后基于每个采样集训练一个基学习器，然后将这些学习器的预测通过投票或者平均的方法进行结合。

随机森林

随机森林在以决策树为基学习器的Bagging的基础上引入了随机属性选择。具体地，决策树在选择划分属性时是选择当前属性集中的最优属性，而在随机森林中对基学习器中每个节点先从属性集合中随机抽取一个包含K个属性的子集然后在这个子集中选择最优属性用于划分数据集。

2.2.2 极端随机树

该算法与随机森林有两点主要的区别：

1、随机森林应用的是Bagging模型用随机采样到的采样集训练基学习器，而极端随机树是使用所有的训练样本得到每棵决策树，也就是每棵决策树应用的是相同的全部训练样本；

2、随机森林是在一个特征集中随机抽取的子集内得到最佳划分属性，而极端随机树是在当前最好的几个（可以指定）划分属性中随机选取一个作为划分属性。

2.2.3 梯度提升 Gradient Boosting

如果说线性回归是台丰田，那么梯度提升就是UH-60黑鹰直升机，这里以一个简单的例子开始。通过一个人是否玩游戏，是否喜欢园艺，是否喜欢戴帽子来预测一个人的年龄，策略是最小化平方损失。我们有9个训练样本并用它们来建模。

PersonID	Age	LikesGardening	PlaysVideoGames	LikesHats
1	13	FALSE	TRUE	TRUE
2	14	FALSE	TRUE	FALSE
3	15	FALSE	TRUE	FALSE
4	25	TRUE	TRUE	TRUE
5	35	FALSE	TRUE	TRUE
6	49	TRUE	FALSE	FALSE
7	68	TRUE	TRUE	TRUE
8	71	TRUE	FALSE	FALSE
9	73	TRUE	FALSE	TRUE

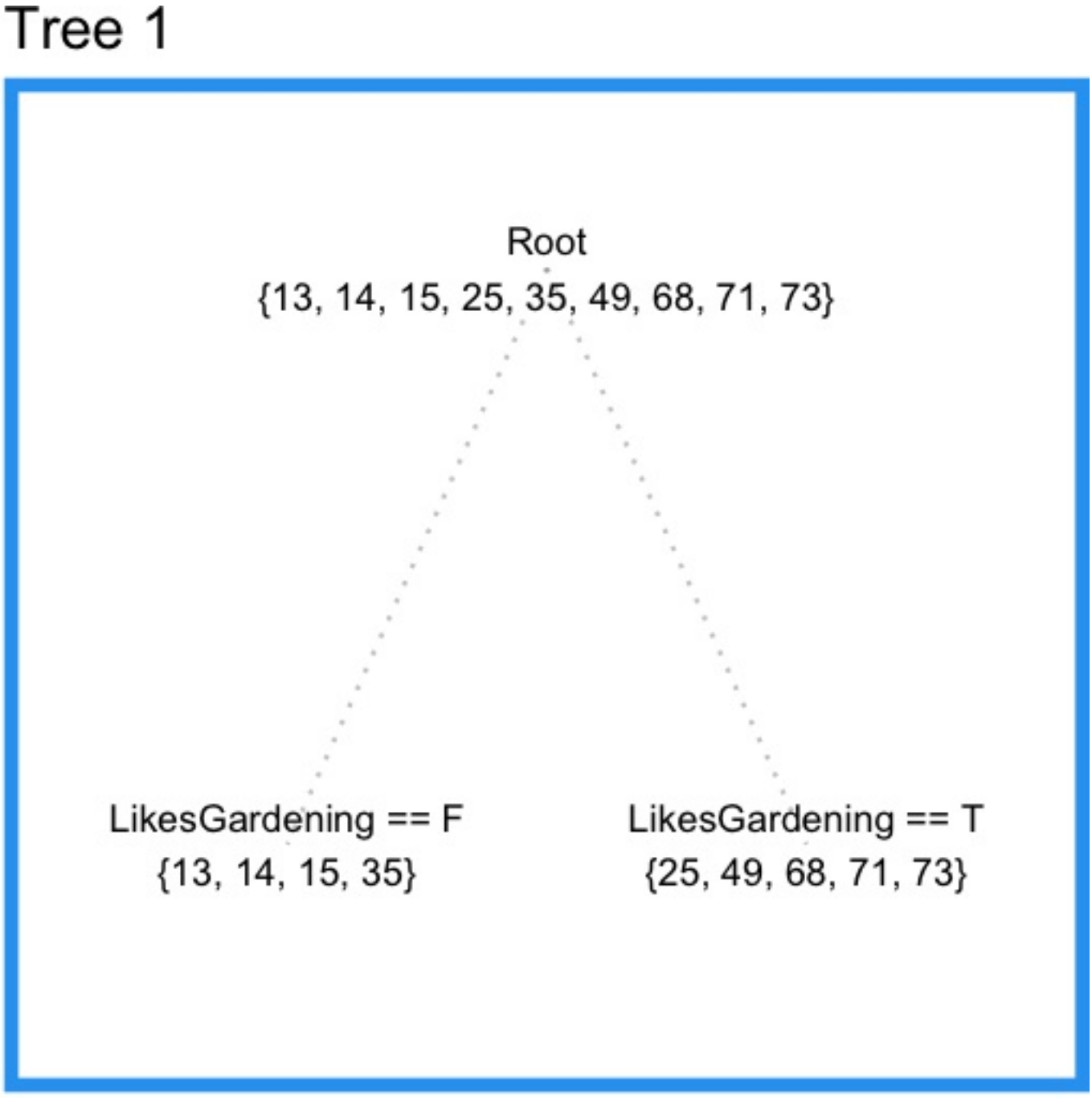
我们凭直觉可以会猜测：

1. 喜欢园艺的更可能是年纪比较大的人
2. 爱玩游戏的的人应该更年轻
3. 喜欢戴帽子可以和年龄没太大关系

看一下数据，大致的知道我们的猜测是否正确

Feature	FALSE	TRUE
LikesGardening	{13, 14, 15, 35}	{25, 49, 68, 71, 73}
PlaysVideoGames	{49, 71, 73}	{13, 14, 15, 25, 35, 68}
LikesHats	{14, 15, 49, 71}	{13, 25, 35, 68, 73}

我们按照是否喜欢园艺划分，得到树1

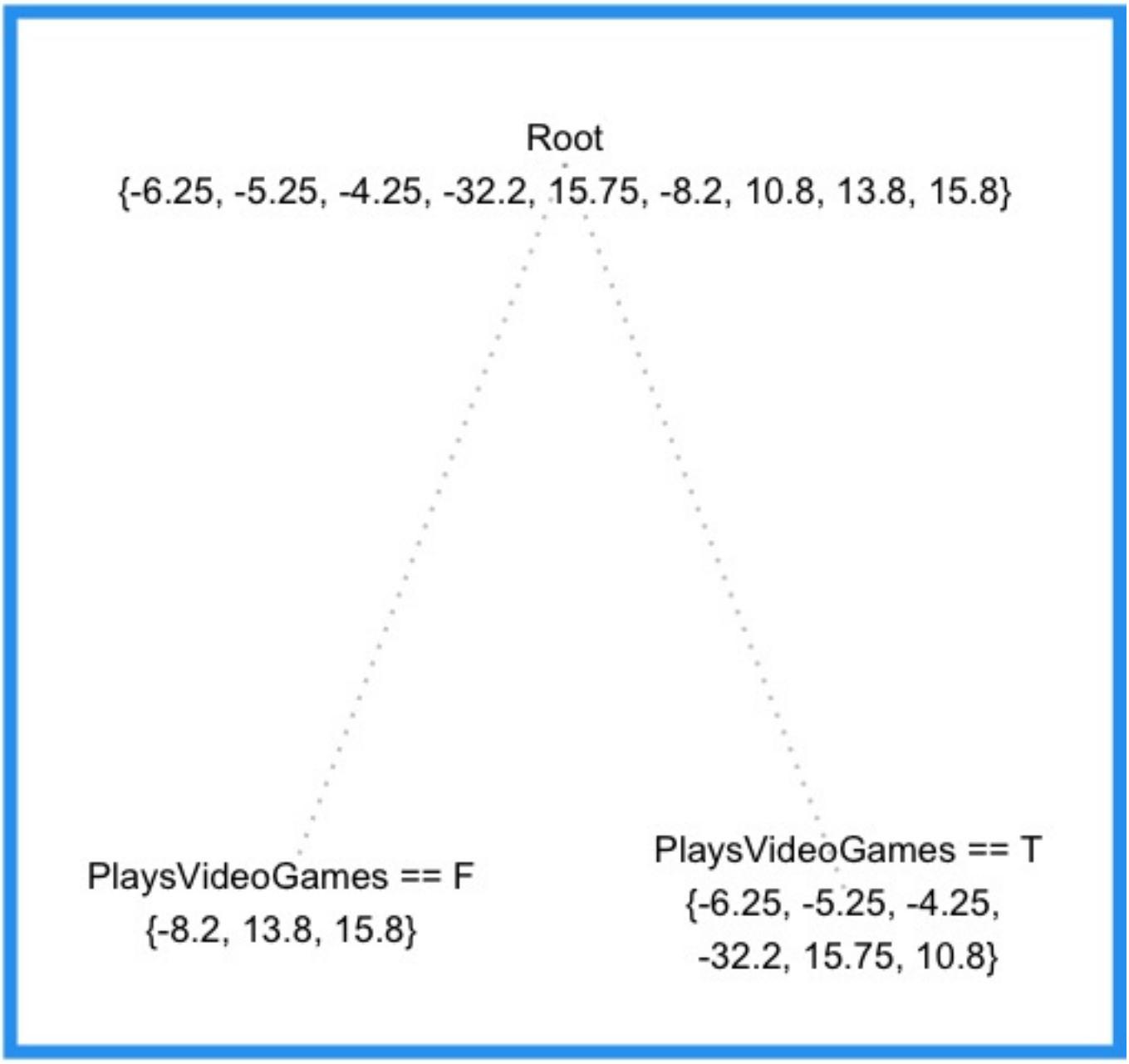


使用树1预测年龄其训练误差如下表

PersonID	Age	Tree1 Prediction	Tree1 Residual
1	13	19.25	-6.25
2	14	19.25	-5.25
3	15	19.25	-4.25
4	25	57.2	-32.2
5	35	19.25	15.75
6	49	57.2	-8.2
7	68	57.2	10.8
8	71	57.2	13.8
9	73	57.2	15.8

我们用树2来对误差数据建模

Tree2



现在我们可以通过误差矫正提高树1的预测

PersonID	Age	Tree1 Prediction	Tree1 Residual	Tree2 Prediction	Combined Prediction	Final Residual
1	13	19.25	-6.25	-3.567	15.68	2.683
2	14	19.25	-5.25	-3.567	15.68	1.683
3	15	19.25	-4.25	-3.567	15.68	0.6833
4	25	57.2	-32.2	-3.567	53.63	28.63
5	35	19.25	15.75	-3.567	15.68	-19.32
6	49	57.2	-8.2	7.133	64.33	15.33
7	68	57.2	10.8	-3.567	53.63	-14.37
8	71	57.2	13.8	7.133	64.33	-6.667
9	73	57.2	15.8	7.133	64.33	-8.667

Tree1 SSE	Combined SSE
1994	1765

Gradient Boosting草稿1

受到上面提升树预测能力的启发，我们得到了梯度提升的原型：

1. 用一个模型拟合训练数据F1(X) = y
2. 用一个模型拟合残差h1(x) = y - F1(X)
3. 得到新模型 F2(x) = F1(x) + h1(x)

通过重复上面步骤，插入更多模型来矫正前一模型的的误差

$$F(x) = F_1(x) \mapsto F_2(x) = F_1(x) + h_1(x) \ldots \mapsto F_M(x) = F_{M-1}(x) + h_{M-1}(x)$$

Gradient Boosting草稿2

现在我们初始化一个只有单一预测值的模型，因为我们的目标是最小化平方损失，我们用目标的均值作为初始模型F0

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) = \arg\min_{\gamma} \sum_{i=1}^{n} (\gamma - y_i)^2 = \frac{1}{n} \sum_{i=1}^{n} y_i.$$

然后像前面一样更新模型 $F_{m+1}(x) = F_m(x) + h_m(x) = y$

Gradient Boosting草稿3

当损失函数为平方损失和指数损失时，每一步的优化是很简单的，但是对于其他的损失函数而言，每一步的优化并不是那么容易，针对这一问题，Freidman提出了梯度提升算法，所用当前模型负梯度作为残差的近似值

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n.$$

Gradient Boosting算法：

1. 初始化产生常量的F0 $F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$
2. 对于m = 1,2...,M，计算伪残差

   $$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n.$$

   - 对于伪残差拟合一个基学习器h(x)
   - 计算每一步的乘数γ_m
   - 更新Fm $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$

2.3 基准模型

使用逻辑回归作为基准模型．逻辑回归经过交叉验证调节参数后表现如下：

Submission and Description	Private Score	Public Score
benchmark.csv a few seconds ago by 夏强	0.89478	0.88803

逻辑回归基准模型的作用：用于判断某个模型是否值得提高性能

III. 方法

(大概 3-5 页)

3.1 数据预处理

3.1.1 特征提取

对于每个出价人选取特征如下

特征提取：

1. 提取特征支付帐号是否与地址相符, 出价数, 参与拍卖数, 每场拍卖出价次数的中位数,
2. 用户国家，Ip，url，登陆设备数及用户的出价数在国家，ip，url，设备上分布的熵值，用户每场拍卖登陆的国家，ip，url，设备数的中位数，平均值和最大值
3. 拍卖最后30分钟的出价数，及拍卖最后30分钟拍卖数占总出价数的比率
4. 出价时间距离拍卖开始结束时间的中位数，同一场拍卖中出价时间与前面其他人出价的间隔的最小值，标准差与平均数与自己的上一次出价的时间间隔的最小值，标准差与中位数

3.1.2 特征工程

1. 缺失值 与出价相关次数的特征均有99个缺失值说明，有99个用户没有进行出价，发现有99人没有出价，故其出价数，ip，url数等特征实际上为0故用0填充，拍卖最后半小时出价数以及出价数占本场出价总数也用0填充，其他缺失用该特征的均值填充。

2. 对于特征采取对数运算np.log10(x+1)以使数据的分布更接近正太分布

3. 对数据进行归一化处理使样本向量转化为“单位向量”，方便后面的运算

4. 类别不均衡问题采用对小类过采样的方法处理

3.2 执行过程

3.2.1 前期探索

前期由于没有提取时间间隔，登陆ip,url,设备等数据，使用随机森林对测试数据进行预测结果如下

后参考讨论帖子：https://www.kaggle.com/c/facebook-recruiting-iv-human-or-bot/discussion/14628

增加上述特征

3.2.2 基准模型

基准模型提交后得分

在这一部分，你需要描述你所建立的模型在给定数据上执行过程。模型的执行过程，以及过程中遇到的困难的描述应该清晰明了地记录和描述。需要考虑的问题：

- *你所用到的算法和技术执行的方式是否清晰记录了？*
- *在运用上面所提及的技术及指标的执行过程中是否遇到了困难，是否需要作出改动来得到想要的结果？*
- *是否有需要记录解释的代码片段(例如复杂的函数）？*

3.3 完善

3.3.1 随机森林

使用sklearn GridSearchCV　交叉验证选择参数

- n_estimators也就是用于集成的基学习器的数量,发现n_estimators在350的时候模型在验证集上有最大的roc_auc分数0.997696060963
- max_features 随机森林的单个树抽取的划分属性的个数在max_features＝3的时候模型在验证集上有最高roc_auc分数0.997721416628
- 使用上述参数预测得分

3.3.2 极端随机树

- n_estimators在950的时候模型在验证集上有最大的roc_auc分数0.998541021354
- max_features＝17的时候模型在验证集上有最高0.998679449576
- 使用上述参数预测得分

3.3.3 梯度提升树

梯度提升树的优化参考：
：https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

- step1:选择一个比较大的学习速率(比如0.1)调节n_estimators，得到最佳n_estimators为760
- step2： 调节基学习器的参数 max_depth，得到基决策树最大深度为12时模型在验证集上有最高roc_auc分数·
- Step 3: 调 gamma　也就是决策树分叉要求的最小能减小的损失，得

到gamma = 0 最佳

- Step 4: 调节subsample（每颗树随机抽取多少样本来学习，默认１也就是使用所有样本） and colsample_bytree（每颗树随机抽取的分割属性，默认１也就是所以所有属性），得到最佳属性 subsample＝0.8, colsample_bytree＝0.7
- Step 5: 调节L1正则项系数 reg_alpha，L2正则项系数reg_lambda　得到最佳参数reg_alpha＝ 1e-20，　reg_lambda＝1e-05
- Step 6: 上述参数选好后优化学习速率，得到最佳学习速率 0.042105263157894736

优化后的模型表现(XGB2.csv)：

Submission and Description	Private Score	Public Score
XGB2.csv a few seconds ago by 夏强 *add submission details*	0.93466	0.90740
ETC.csv 18 hours ago by 夏强 *add submission details*	0.92307	0.89891
RFC.csv 19 hours ago by 夏强 *add submission details*	0.92886	0.90917
benchmark.csv a day ago by 夏强	0.89478	0.88803

3.3.4 结合前面三种模型

使用这三种模型的预测结果的均值作为预测结果，结果如下

Submission and Description	Private Score	Public Score
combined.csv a few seconds ago by 夏强	0.93012	0.90922

3.3.5 最终模型

pulic score 随机森林模型和三种模型组合的得分最高均为0.909但是随机森林模型比组合模型要简单很多，故最终采用的模型为随机森林

3.3.6 其他尝试 stacking

采用双层结构，用上述模型及基准模型对数据的预测值作为下一层的输入，拟合一个梯度提升树，然后对结果进行预测，预测结果如下

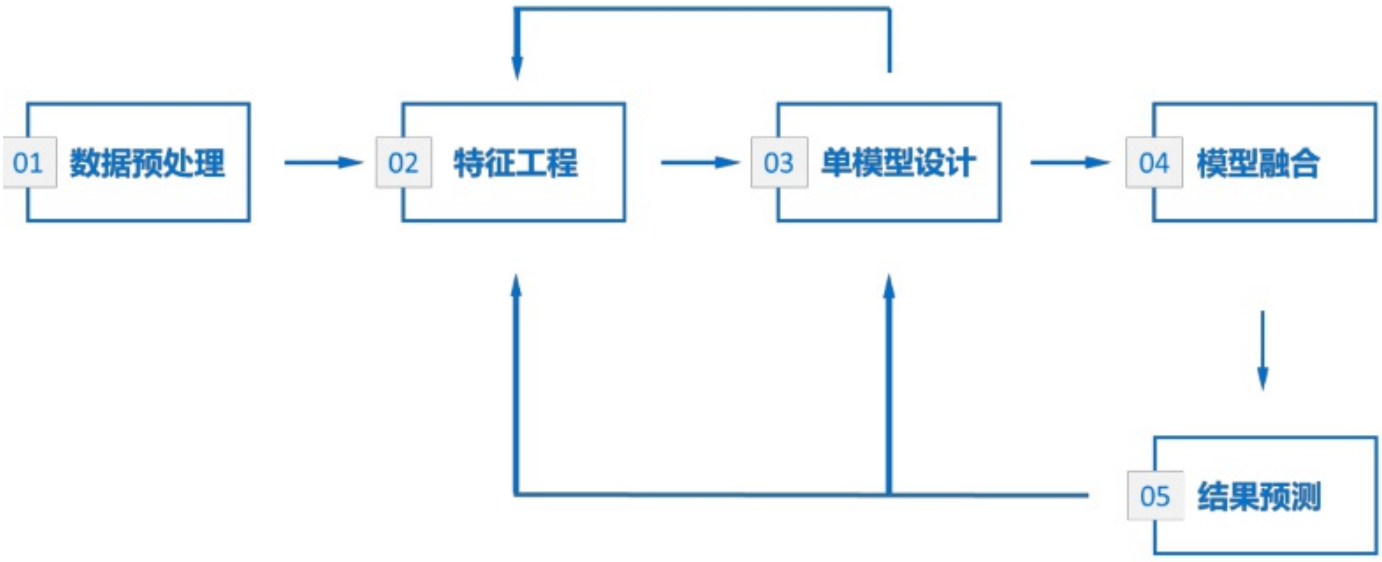
stacking.csv 6 days ago by 夏强 *add submission details*	0.66933	0.58017

分数很低，低于基准模型，放弃之·

结论

总结

流程总结：



这个项目中比较难的地方有：

1. 没有相关拍卖经验，对于提取的特征那些重要那些不重要没有一个认识，需要不断尝试
2. 有些算法如梯度提升树参数众多找到最优参数组合非常困难，只能一个一个调参数·

一个调节很多参数算法的数字型参数的点子步骤如下：

- 设置一个阀值gamma如果分数提高小于它就不调
- 对于所有参数设置初始值，值的上界和下界在上下界间均匀选择候选值
- 循环的优化每个参数，选出新的最优参数更新上下界，候选值直到优化参数不提高参数或者提高的参数低于阀值

将来可能的改进

- 随机森林算法中每颗树的参数有一定的优化空间
- 梯度提升树的参数可能还可以进一步优化
- 增加或减少一下特征

参考文献

roc曲线： https://zh.wikipedia.org/wiki/ROC%E6%9B%B2%E7%BA%BF

roc曲线下面积计算方法总结：http://blog.csdn.net/pzy20062141/article/details/48711355

什么是特征工程： https://www.zhihu.com/question/29316149

《统计学习方法》，李航，第5章决策树，第8章提升方法 《机器学习》，周志华，第8章集成学习 Random_forest wiki: https://en.wikipedia.org/wiki/Random_forest

xgboost文档关于提升树的介绍

http://xgboost.readthedocs.io/en/latest/model.html

梯度提升的解释： http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting/