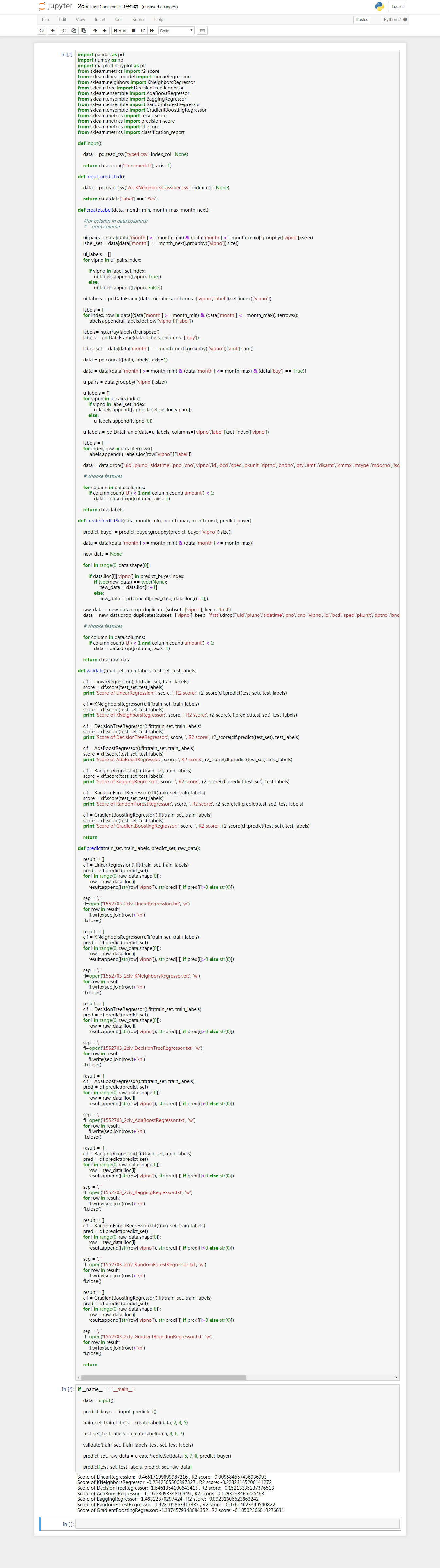
## 代码运行

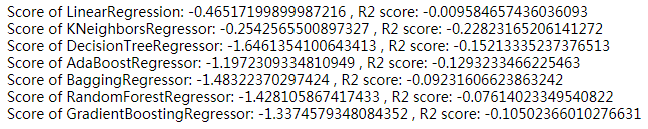
[点击查看原图](civ.png)

运行所需type4.csv文件可在a的特征结果中找到



### 讨论分析

1. 本小题针对用户在第k+1个月存在购买行为，以用户购买商品总体耗费金额作为回归问题的目标值；构建用户交易特征与用户购买金额的机器学习回归模型，重复上述预测
2. 本小题中使用了回归器取代了分类器，而其中朴素贝叶斯算法我选择替换为了线性回归算法
3. 本题中选择使用R2\_score进行结果的评估，训练集则使用包含“amount”和“count” 的列进行预测
4. 以下是结果截图



* 1. 从r2\_score的定义来说，完全正确是1，使用全部训练数据的平均值进行预测得到的0，因为r2的计算方式为1-u/v，而这里七个模型获得的结果评分均小于0，其意义代表差距极大，甚至不如使用平均值作为统一的预测结果。
  2. 回头去看各回归器的原理，是模拟一个函数曲线去拟近训练集数据构建的曲线
  3. 再回头看小题中获得的结果，不论是按照vipno或者bndno或者其他列作为横轴，都难以观察到一个规律的函数图像
  4. 分析后认为，对于回归器而言
     1. 可连续的数据效果好于完全离散的数据
     2. 维数过多的特征集对回归器而言可能是噪声的集合，不一定能获得一个好的结果
  5. 尝试减少特征数，但是发现针对单个用户的购买记录实际不多，当特征数减少后，依然保持完全离散的状态，不足以构建一条值得模拟的函数曲线，并且产生了大量的随机性，导致各回归器之间的差异变大，r2指数高的回归器再迭代一次之后进行验证，指数大幅降低，认为是偶然性导致的指数变高，故不采用。
  6. 最后选择保留最初的预测结果，虽然它的r2\_score小于0，但是具有一定的普遍性
  7. 对预测结果进行深入分析（此处未截图），预测结果在不同列作为x轴进行绘图的情况，都存在尽量拟合实际标签的结果，但是相反的部分可能更多。总体预测结果范围可控，不存在预测结果过大的情况。此外若预测结果为负值，则将其修正为0，使其符合实际并方便下一小题的使用

### 性能比较

下图比较了几个回归器的性能，耗时包括一次训练和一次预测