Task2: Note

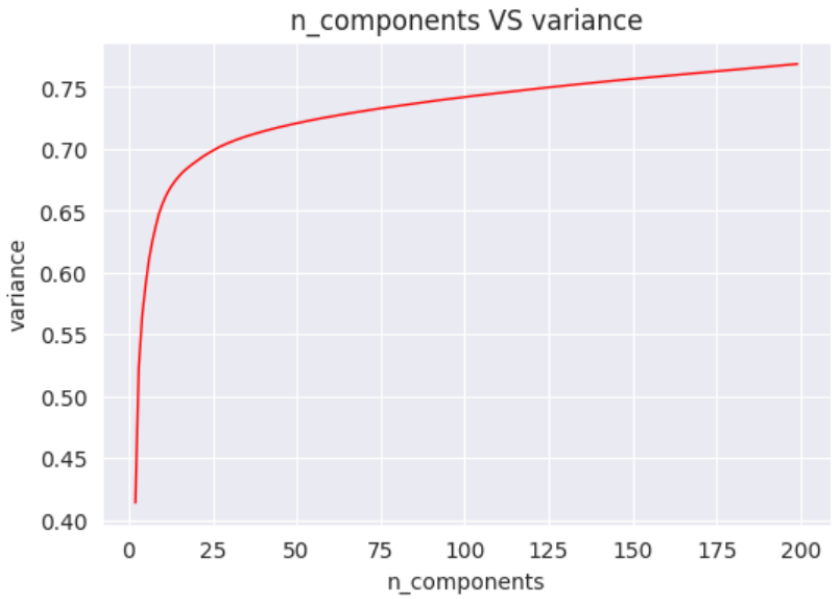
在task1中，我只尝试了前20000的特征，所以在task2中我想尝试更多的特征。我一开始选择了100000特征，但我发现处理起来内存不够太少了。我就在想如何降维才能让我能读取所有的特征，但又不超过我的内存的负荷。

PCA

首先我考虑的是使用算法进行降维，我首先读取了100000条特征，然后定义n_components为2-200，并且画出n_components和explained_varianec的折线图，尝试找到肘部。最后我发现肘部是15-25的区间。

```
108]: plt.figure(figsize=(6,4))
      plt.plot([i for i in range(2, 200)], variances, color="red", linewidth=1 )
      plt.xlabel("n_components")
      plt.ylabel("variance")
      plt.title("n_components VS variance")
```

```
108]: Text(0.5, 1.0, 'n_components VS variance')
```



我选择了15作为n_components，并对降维后的数据进行调参，但最后结果并不好

日期: 2023-08-17 21:41:10

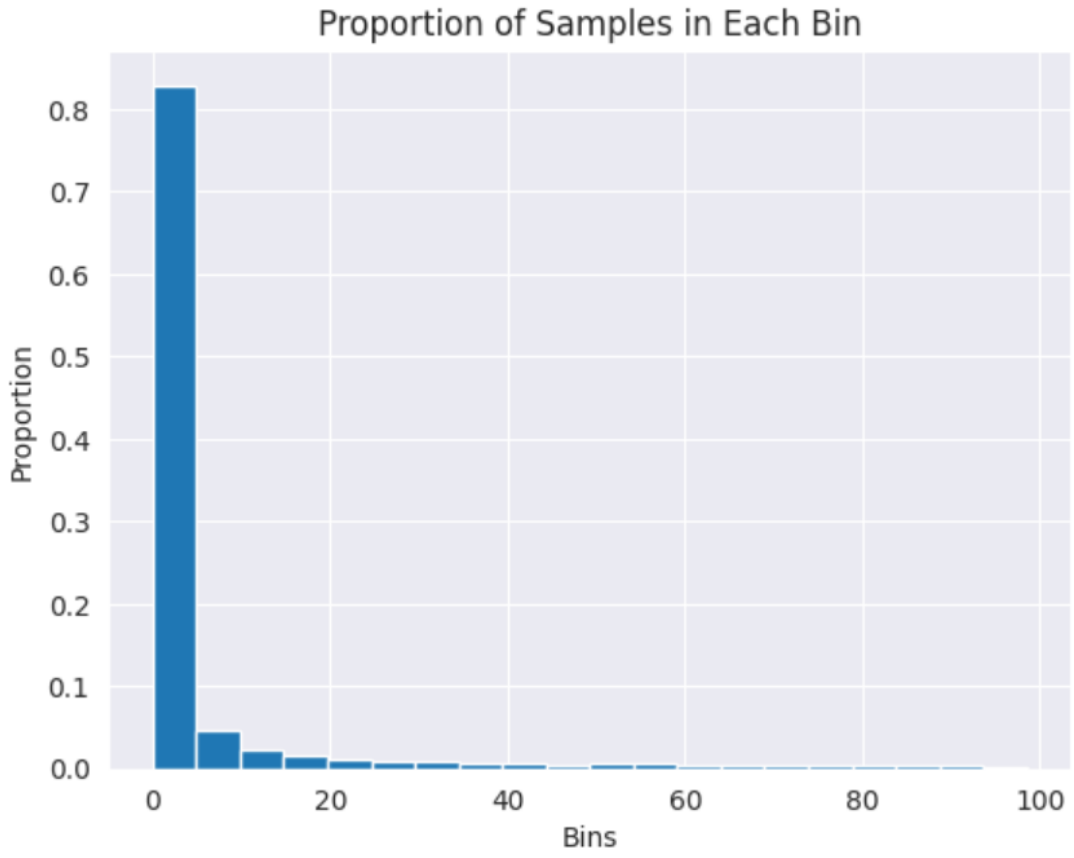
score: 19.6319

mae_control: 22.0453

mae_case: 17.2184

缺失值

同样的，为了减少无效特征，我计算了所有特征的缺失值占比，我发现有很多特征的缺失值占比百分之60以上，为了进一步提高效率，我删掉了这部分特征



重要性特征

最后，我使用了baseline(LightGBM)模型训练这缺失值占比百分之四十以下的特征，使用lightGBM的函数.feature_importances_得到了特征重要性值。我删掉了重要性为0的特征，最后只剩下5000+特征（从100000特征中）。我计算了模型在这两个数据中训练后预测的mean absolute error，我发现5000+的特征相比于100000的特征mae只上升了0.2 (0.7 → 0.9)，我意识到这个方法的可行性。

最后

我把480000+个特征，划分成五组，并且假设只有组内的特征有相关性，组与组之间相互独立。然后我通过baseline计算出重要性非0的特征并且保存。最后我得到了一个数据集，有20000+的特征，但每个特征都是对预测年龄很重要的甲基化数据。之后我的baseline得分是

经过调参达到2.2

future

之后计划

1. 多训练几个模型，并ensemble这些模型看看效果，缓和过拟合
2. 通过特征工程进一步减少损失