员工离职预测实验报告

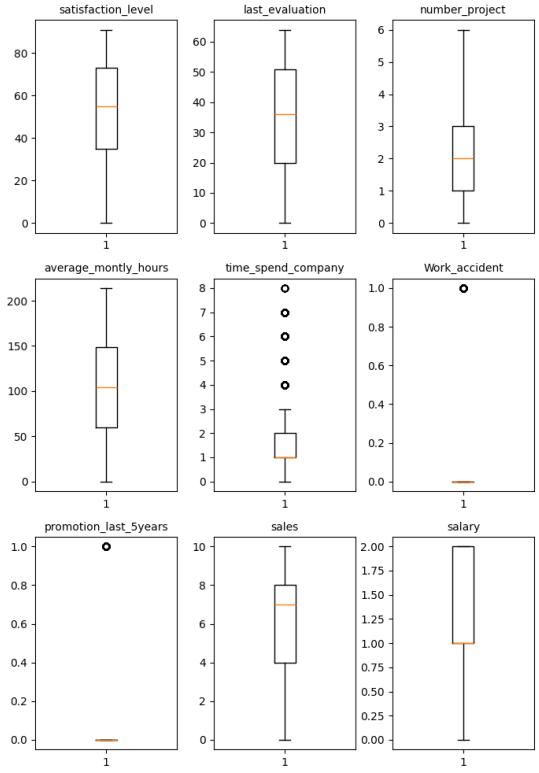
杨坤泽
2023210799

2023 年 11 月 29 日

1 数据处理

先分析实验数据，对数据进行预处理，提取特征和标签以及对特征的文字数据进行编码。其中的编码方式依旧利用 sklearn 库中的 LabelEncoder 实现，之后对所有数据进行归一化和标准化。

为分析实验数据是否存在异常值，利用 boxplot 对所有特征绘制箱型图，结果如下：



不难注意到前四个特征以及后两个特征的训练数据没有异常值。对于 time_spend_company 特征，观察数据发现存在工作年限较长 (10 年左右) 的员工；对于另两个特征则都为 0、1 分布，因此这三个特征其实也并不存在异常值。

观察实验数据，发现标签中的 0 和 1 的数量并不均衡 (1 类样本数量多于 0 类样本)，考虑采用集成方法，利用多个对 1 类欠采样和所有 0 类构成的样本训练得到的子模型求期望得到最终的训练模型。在代码实现中采用随机采样的方法，在每个 loop 循环中对 0 类完成欠采样。

2 随机森林算法实现

采用随机森林方法，定义 random_forest 类，主要的算法与训练实现定义为 fit() 函数，如下：

```
def fit(self, x, y):
    tbar = tqdm(range(self.num))
    idx = []
    for _ in tbar:
        tbar.set_description('Training Sub Decision Tree into Random Forest'
                                                )
        tree = DTC(max_depth = self.depth, random_state = self.seed,
                                                max_features = 'sqrt')
        sample_index = self.rd.randint(0, x.shape[0], self.sample)
        tree.fit(x[sample_index], y[sample_index])
        self.trees.append(tree)
        idx.append(sample_index)
    self.idx = np.array([i for i in range(x.shape[0]) if i not in np.unique(
                                            np.array(idx))])
```

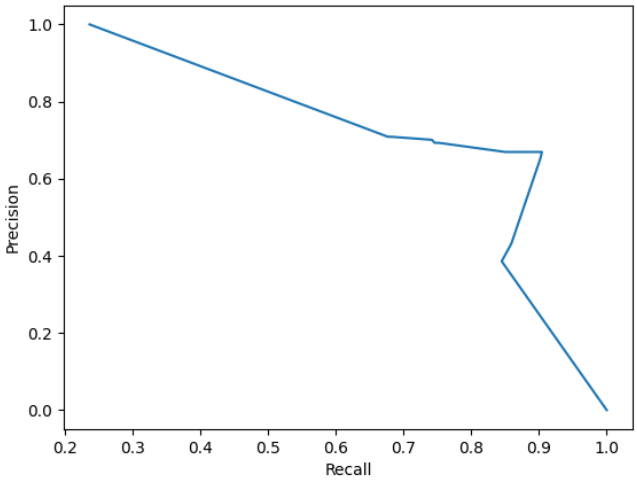
利用 DecisionTreeClassifier 实现森林中的每一课决策树，通过 max_features 实现特征的随机选取，以及通过对数据的自取采样实现每棵树的训练数据集的构建。这里需要记录森林中每棵树训练数据的下标，方便在最后评估模型性能时计算 out of bag error。对于 PRC 曲线的绘制，利用 sklearn.metrics 里的 precision_recall_curve 函数，通过输入真实标签与预测概率实现。

3 实验结果

对于不采用欠采样进行数据均衡的随机森林方法，各项参数设置为森林中的树的数量 tree_num = 100、每棵树的最大深度 max_depth = 2、自取采样数 sample_num = 500 等。最终模型的 OOB 准确率为：

```
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1216.41it/s]
Oob-accuracy without data balance method:  0.9052044609665427
```

可见模型对于包外数据实现了较好的分类准确率。模型的 PRC 曲线如下：



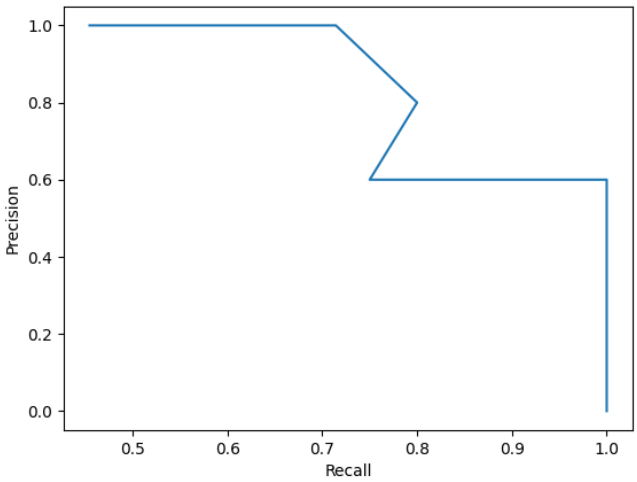
注意到曲线整体较为偏向右上侧，但趋势并不明显，说明模型的数据存在较为不均衡的现象。

对于采用欠采样进行数据均衡的随机森林方法，各项参数设置同上，经过 10 轮训练，模型的平均准确率为：

```
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1181.55it/s]
Oob-accuracy with data balance method for 0-th model:  0.8181818181818182
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1163.68it/s]
Oob-accuracy with data balance method for 1-th model:  0.9090909090909091
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1216.73it/s]
Oob-accuracy with data balance method for 2-th model:  0.8181818181818182
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1202.17it/s]
Oob-accuracy with data balance method for 3-th model:  0.7272727272727273
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1177.25it/s]
Oob-accuracy with data balance method for 4-th model:  1.0
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1202.24it/s]
Oob-accuracy with data balance method for 5-th model:  1.0
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1195.77it/s]
Oob-accuracy with data balance method for 6-th model:  1.0
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1185.81it/s]
Oob-accuracy with data balance method for 7-th model:  0.8181818181818182
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1205.16it/s]
Oob-accuracy with data balance method for 8-th model:  0.9090909090909091
Training Sub Decision Tree into Random Forest: 100%| 100/100 [00:00<00:00, 1230.76it/s]
Oob-accuracy with data balance method for 9-th model:  0.8181818181818182
Oob-accuracy with data balance method:  0.8818181818181818
```

注意到模型的分类准确率有所降低，这是因为数据相对而言有缺失，特征信息并不充分，可能会导致分类错误的结果。其中一个子模型的 PRC 曲线如下：

在实验过程中发现采用数据均衡的方法后，训练得到的子模型都更偏向右上侧。说明模型的 PRC 曲线更优，即训练数据更为均衡，虽然分类准



确率有所下降，但得到的模型仍是“好”的分类器。