Lab3 Report

计11 周韧平 2021010699

算法说明

Bagging (Bootstrap Aggregating)

假设我们有一个训练集 (X, Y)，其中 X 是输入特征矩阵，Y 是目标变量向量。我们的目标是训练 M 个基学习器。其训练过程可以形式化的定义为：

1. **Bootstrap 抽样**：对于每个基学习器 (m = 1, 2, ..., M)，从训练集中进行 Bootstrap 抽样，即**有放回地**从 X 和 Y 中抽取样本。这样每个基学习器都有可能包含训练集的不同子集。
2. **训练基学习器**：使用 Bootstrap 抽样得到的样本训练基学习器 h_m。
3. **集成预测**：对于分类问题，通过投票进行预测。对于回归问题，取平均预测值。

AdaBoost

AdaBoost 通过反复修改数据的权重并重新训练基学习器，以使得后续的基学习器集中在前一个学习器错分的样本上，从而提高模型的性能。

假设我们有一个训练集 (X, Y)，其中 X 是输入特征矩阵，Y 是目标变量向量。我们的目标是训练 M 个基学习器。其训练过程可以形式化的定义为：

初始化样本权重：将每个样本的权重初始化为相等的值。

训练基学习器：对于每个基学习器 (m = 1, 2, ..., M)：

i. 使用当前样本权重训练基学习器 h_m。

ii. 计算基学习器在训练集上的错误率 $error$。

iii. 计算基学习器的权重 $\alpha_m = 0.5ln(\frac{1-error}{error})$，表示该学习器在最终预测中的重要性。

iv. 更新样本权重，使得错分样本的权重增加，正确分类的样本权重减少。更新公式正例为 $W_i = W_i exp(-\alpha)$，反例 $W_i = W_i exp(\alpha)$

集成预测：对于每个样本，根据基学习器的权重加权预测结果，最终的预测结果是根据 α 加权投票的结果。

但在基于分类的 AdaBoost 模型要求分类任务是二分的，因此不能直接将算法用到当前问题上，我采取了两种方法解决该问题：1.将评分对应的五分类问题拆分成二分类问题，并采用 One-vs-All 的方式选择最终的分类结果。2.使用基于回的AdaBoost算法，其区别在于错误率计算公式为 $error_i = \frac{|\hat{Y}_i - Y_i|}{max(|\hat{Y}_i - Y_i|)}$

实验

测量指标

本次实验中我选的测量指标如下：

- **MAE (Mean Absolute Error)**: 平均绝对误差，是预测值与真实值之间的绝对误差的平均值。它的计算方法是将每个预测值与对应的真实值相减取绝对值，然后求所有绝对值的平均值。

- **RMSE (Root Mean Squared Error)**: 均方根误差，是预测值与真实值之间误差的平方的平均值的平方根。它的计算方法是将每个预测值与对应的真实值相减得到误差，然后求这些误差的平方的平均值，最后取平方根。
- **Accuracy (准确率)**: 分类任务中最常用的评价指标之一，表示模型预测正确的样本所占的比例。计算方法是将模型正确分类的样本数除以总样本数。
- **MAPE (Mean Absolute Percentage Error)**: 平均绝对百分比误差，是预测值与真实值之间的百分比误差的平均值。计算方法是将每个预测值与对应的真实值相减得到误差，然后将这些误差除以真实值，再取绝对值，最后求所有绝对值的平均值。
- **Precision (精确率)**: 真正例（模型将其判定为正类且实际为正类的样本）的数量除以模型将其判定为正类的样本的数量。多分类则分别假设每个类别为正例计算平均值
- **Recall (召回率)**: 将真正例的数量除以实际为正类的样本的数量。多分类则分别假设每个类别为正例计算平均值
- **F1 Score**: 综合考虑了精确率和召回率的指标，是精确率和召回率的调和平均值。F1 Score的计算方法是先计算精确率和召回率，然后利用这两个值计算F1分数，公式为：$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$

数据预处理和特征处理

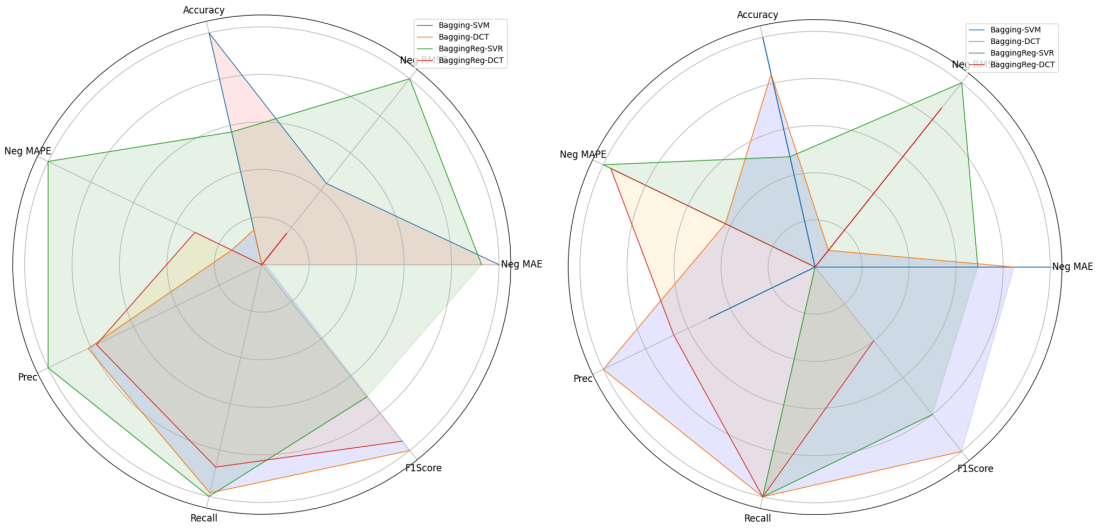
实验中我主要对 reviewText 这一个键的内容做了特征提取，去除停用词后，采用了词袋模型和TF-IDF向量的方式来提取特征。我采用 9：1的比例分割了数据集和测试集，为了提升训练效率，我从训练集中采样了一部分来训练，并为特征维度设置了上限。

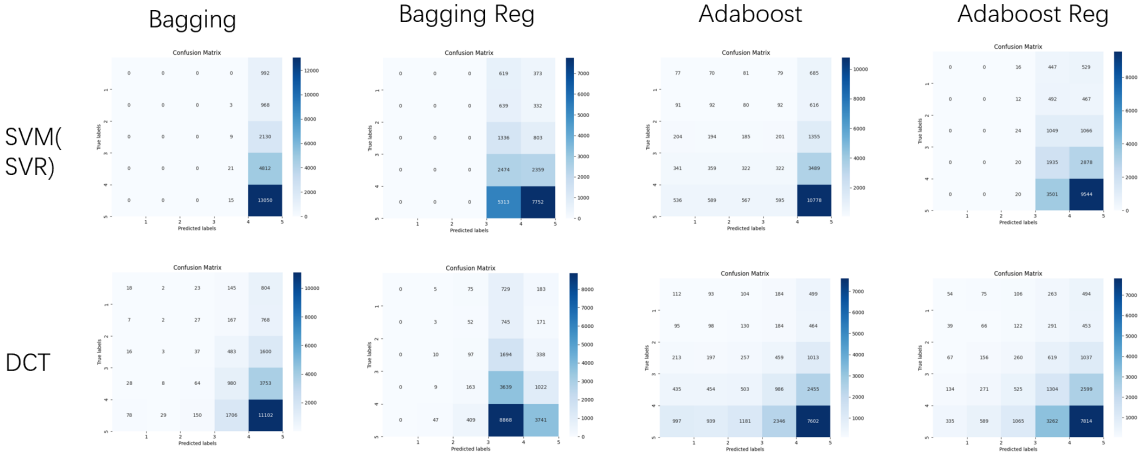
对比集成方法时，我对比了Adaboost或Bag+SVM或DCT，共四种组合，对于其中每种组合，我还对比了其采用回归和分类模型的效果。而在其他实验中，我选取了Adaboost在回归方法下的模型和Bagging在分类方法下的模型

两种集成方法对比

我在学习器数量为10，训练集抽样1%，特征维度为100的TF-IDF特征向量下比较了四种Bagging模型和Adaboost模型的性能，下面的所有实验中，如果没有特殊说明，统一采用此设定。此外，我还绘制了其雷达图与对应的混淆矩阵，为了让雷达图能更好地展示模型性能，我将在绘制雷达图时将所有越小越好的值去了负，并对模型做了归一化处理

Method-Base_Model	MAE	RMSE	Accuracy	MAPE(%)	Prec	Recall	F1Score
Adaboost-SVM	0.91	1.50	0.52	36.35	0.23	**0.23**	0.21
Adaboost-DCT	1.11	1.66	0.41	38.51	0.22	0.22	0.21
AdaboostReg-SVR	0.74	1.12	0.52	33.17	0.23	0.22	0.20
AdaboostReg-DCT	0.94	1.40	0.43	35.54	0.22	0.22	**0.22**
Bagging-SVM	**0.73**	1.31	**0.59**	36.6	0.21	0.2	0.15
Bagging-DCT	0.75	1.29	0.55	35.16	**0.24**	0.22	0.2
BaggingReg-SVR	0.77	**1.09**	0.46	**33.2**	0.18	0.22	0.19
BaggingReg-DCT	0.86	1.12	0.34	33.32	0.22	0.22	0.17



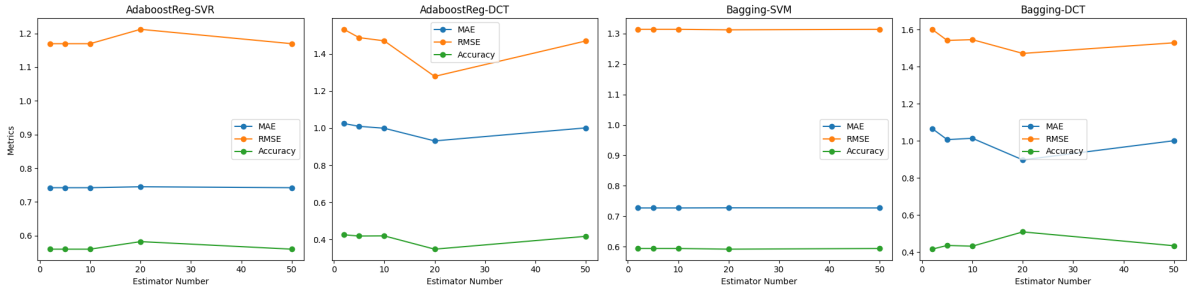


分析：

- **从基分类器的角度来看，无论采用哪种集成学习方法，支持向量机的能力都要强于决策树**，可以看到雷达图中，Bagging-SVR和Adaboost-SVM分别占据榜首，而DCT模型仅仅在 F1 score 上稍有优势。
- 通过观察混淆矩阵，我们可以发现，**支持向量机比决策树更容易陷入模式崩塌的情况**，即将所有样本均预测为5.00，而通过观察数据集可以发现，数据集中有超过一半的标签均为5.00，因此可以合理猜测，支持向量机性能上的优势可能正是因为其比决策树更容易模式崩塌导致的。而DCT由于不容易模式崩塌，即使在面对不均衡的样本时也可以保持相对均衡的预测值，因此其在F1 score这种综合考虑了召回率和准确率的综合性能指标上取得了更好的结果
- **不同指标对任务的偏好不同，整体来看，将集成学习视作分类任务往往准确率更高，而视作回归任务则更容易在MAE和RMSE指标上取得优势**，分析其原因可能为，回归任务预测的是一个连续的值，而分类任务则预测的是离散的点，对于MAE，RMSE这样的以“距离”为标准的指标来说，离散的值可能会扩大真值附近预测值和真值之间的误差，因而导致指标较大，而回归任务由于预测的是连续的值，模型并没有完全聚焦到预测一个离散标签的任务上，因此其准确率会低于分类任务。

学习器数量

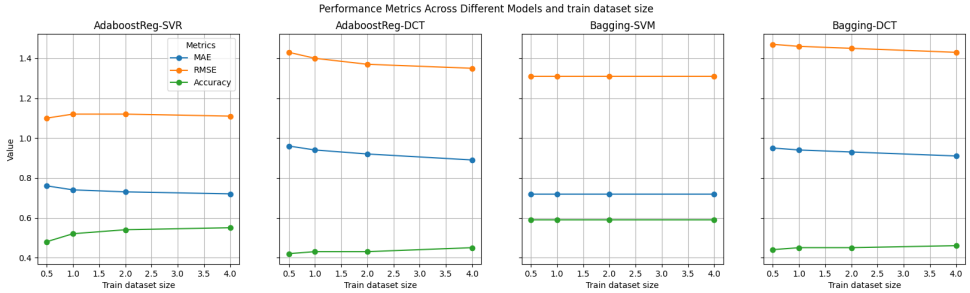
本节中我选用的词袋模型探究学习器数量对模型的影响



实验分析：

- 整体来看，学习器数量增加对模型性能的提升并不明显，特别是对SVM+BAGGING来说，增加学习器数量时模型性能几乎保持不变，从后面的实验也可以看出，Bagging+SVM的组合对于实验超参数的鲁棒性较好
- 对于AdaboostReg+SVR组合来说，在学习器数量达到20时模型性能最好。对于大部分模型来说，学习器数量在20以内即可以取得一个比较好的性能结果。

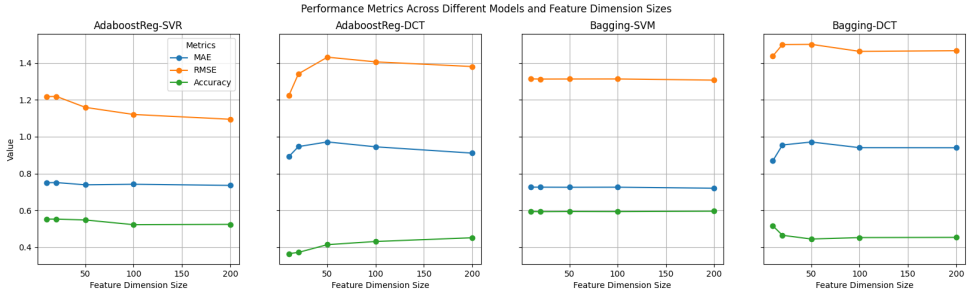
训练集大小



实验分析：

- 整体来看，训练集规模增大，模型性能有一定的提升，特别是准确率指标，在四个模型上都得到了较大幅度的提升
- 决策树相比支持向量机对训练集规模更加敏感，更大的训练集可以为其带来更高的性能提升。这与之前的实验相吻合，由于SVM易于陷入模式崩塌，因此增加训练数据量并不会提升其性能

特征维度大小

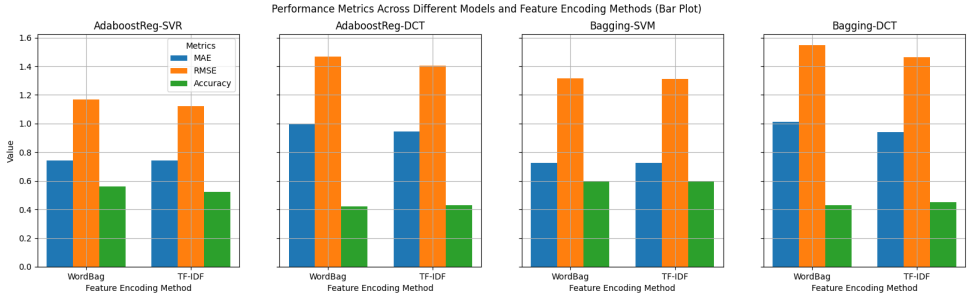


- 整体来看，增加特征维度似乎并没有提升模型性能，这有悖于常理。分析其原因，可能是因为维度增加带来更复杂的特征，而在训练数据量抽样比例不大的情况下，提高模型维度更容易导致过拟合，模型可能会学到特征中的噪音。

- 在Adaboost+DCT组合下，模型性能得到了一定的提升，但仔细观察可以发现，在特征维度较小时，增加维度首先会导致 MAE 和 RASE 变差，然后才逐渐变好，因此可以合理猜测，对于其它模型组合来说，只有当数据维度提升超过某一阈值后，数据维度增加带来的信息增益才会真正体现出来。

特征提取方式

本节我对比了词袋特征和TF-IDF特征对模型性能的影响



实验分析：

- 整体来看，两种特征数据训练出的模型性能差异不大，TF-IDF相比词袋模型有较小幅度的提升，这可能是因为TF-IDF在提取特征时考虑了文档频率和逆文档频率，包含了更多的全局信息，考虑到训练时是对训练集抽样训练，这种包含更多整个训练集特征的数据可以让模型更不容易过拟合
- DCT相比SVM整体性能较差，但TF-IDF带来的模型提升也更为明显，正如之前分析的，DCT不容易模式崩塌，在极不均衡的数据集上这反而成为其劣势，而TF-IDF包含了更多的全局特征信息，对于DCT来说学到的结果也更容易泛化到测试集上。而SVM本身容易模式崩塌，无论如何改变特征，可能都难以阻止模型”摆烂“，因此对性能的提升也就不明显。