Report

实验设计

本实验为实现不同的集成学习算法，并对比它们与不同的基分类器结合时的效果。

实验任务为基于评论的评分预测任务，在本实验中该任务基于回归任务来实现。

本实验在模块 `models.py` 中实现了两个类： `Bagging` 和 `Adaboost` ，分别对应两个集成学习算法。每个类有对应的初始化方法 `__init__` 以及拟合与预测方法： `fit` 与 `predict`

在主程序 `main.py` 中会首先读取数据并将其按照 $9:1$ 的比例划分为训练集和测试集。模型使用全部的训练集和测试集进行训练测试。

读取完数据集，程序会根据参数选取相应的基回归器： `svm.LinearSVR` 和 `tree.DecisionTreeRegressor` （使用 `scikit-learn` 中的工具包），然后根据参数选取相应的集成学习算法类。

之后变调用类中的 `fit` 与 `predict` 方法对训练集进行拟合，然后在测试集上输出预测结果并计算指标。

实验结果

Baseline

使用以下命令运行实验

```
python main.py --regressor svm --ensemble baseline
python main.py --regressor tree --ensemble baseline
```

得到的结果如下：

基回归器	MAE	MSE	RMSE
LinearSVR	0.8181	1.4875	1.2196
DecisionTreeRegressor	0.7500	1.0916	1.0448

Bagging集成

参数选择为 $n = 5, ratio = 0.8$
使用以下命令运行实验

```
python main.py --n 5 --ratio 0.8 --regressor svm --ensemble bagging
python main.py --n 5 --ratio 0.8 --regressor tree --ensemble bagging
```

基回归器	MAE	MSE	RMSE
LinearSVR	0.8112	1.4114	1.1880
DecisionTreeRegressor	0.7323	0.9765	0.9882

参数选择为 $n = 5, ratio = 0.5$
使用以下命令运行实验

```
python main.py --n 5 --ratio 0.5 --regressor svm --ensemble bagging
python main.py --n 5 --ratio 0.5 --regressor tree --ensemble bagging
```

基回归器	MAE	MSE	RMSE
LinearSVR	0.8071	1.3774	1.1736
DecisionTreeRegressor	0.8112	1.4114	1.1880

参数选择为 $n = 10, ratio = 0.8$
使用以下命令运行实验

```
python main.py --n 10 --ratio 0.8 --regressor svm --ensemble bagging
python main.py --n 10 --ratio 0.8 --regressor tree --ensemble bagging
```

基回归器	MAE	MSE	RMSE
LinearSVR	0.7949	1.3510	1.1623
DecisionTreeRegressor	0.7269	0.9561	0.9778

AdaBoost集成

参数选择为 $n = 5$
使用以下命令运行实验

```
python main.py --n 5 --regressor svm --ensemble adaboost
python main.py --n 5 --regressor tree --ensemble adaboost
```

基回归器	MAE	MSE	RMSE
LinearSVR	0.8037	1.4728	1.2136
DecisionTreeRegressor	0.7044	1.0577	1.0284

参数选择为 $n = 10$
使用以下命令运行实验

```
python main.py --n 10 --regressor svm --ensemble adaboost
python main.py --n 10 --regressor tree --ensemble adaboost
```

基回归器	MAE	MSE	RMSE
LinearSVR	0.7990	1.4784	1.2159
DecisionTreeRegressor	0.6841	1.0968	1.0473

实验分析

从实验结果我们可以看出，集成学习算法的确在一定程度上提高了基分类器的性能

比较两个集成学习算法，我们可以看出 Adaboost 在大多数情况下都优于 Bagging ，这可能是因为 Adaboost 在训练过程中会调整样本的权重，使得错误的样本在后续的训练中得到更多的关注，从而提高了模型的泛化能力。

同时，对比不同分类器个数的情况，我们可以看出，一般情况下，分类器个数越多，模型的性能越好，这可能是因为集成学习算法的本质是通过多个弱分类器的组合来构建一个强分类器，因此分类器个数越多，模型的性能越好。

对比Bagging中不同的ratio选择，可以发现，对于svm， $ratio = 0.5$ 的时候性能更好，而对于决策树， $ratio = 0.8$ 的时候性能更好，这可能是因为svm对于数据的拟合能力更强，因此需要更多的随机性来提高模型的泛化能力。

实验讨论

在该实验中我们完成了两个集成学习算法的实现，并对其进行了实验。发现集成学习算法的确在一定程度上提高了基分类器的性能。初次之外，本实验还有一些可以改进的地方：

1. 本实验中我们只使用了两个基分类器，可以尝试使用更多的基分类器来进行实验，看看模型的性能是否会有所提高。
2. 本实验中我们只使用了两个集成学习算法，可以尝试实现更多的集成学习算法，如AdaBoost.M1等，看看模型的性能是否会有所提高。
3. 本实验中我们只使用了一个数据集，可以尝试使用更多的数据集来进行实验，看看模型的性能是否会有所提高。