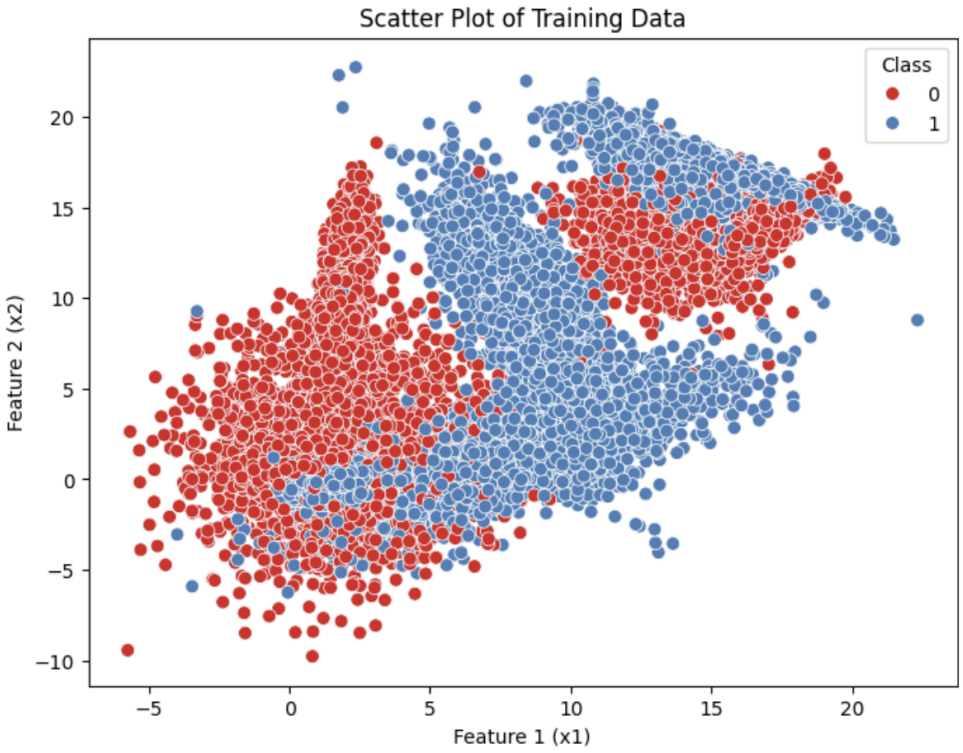
مبانی هوش محاسباتی (پروژه سوم)

کیارش آستان‌بوس ۴۰۰۱۲۶۲۵۷۰
محمدرضا نادری ۴۰۰۱۲۶۲۳۸۶

مهر ۱۴۰۳

فهرست مطالب

۱) Visualize



شکل ۱:

با نمایش داده ها میتوان دید دو کلاس داریم که هر کدام به دو بخش تقسیم شده اند. همچنین تراکم نیز در دو دسته سمت چپ داده ها بیشتر است.

در هر دسته نیز تعداد اندکی از کلاس مقابل وجود دارد که نویز محسوب می‌شود. وهمچنین چندین داده پرت از هر دو کلاس داریم که به تراکمی نزدیک نیستند.

۲) Bagging

۱.۲) Training

: GridSearch Parameters

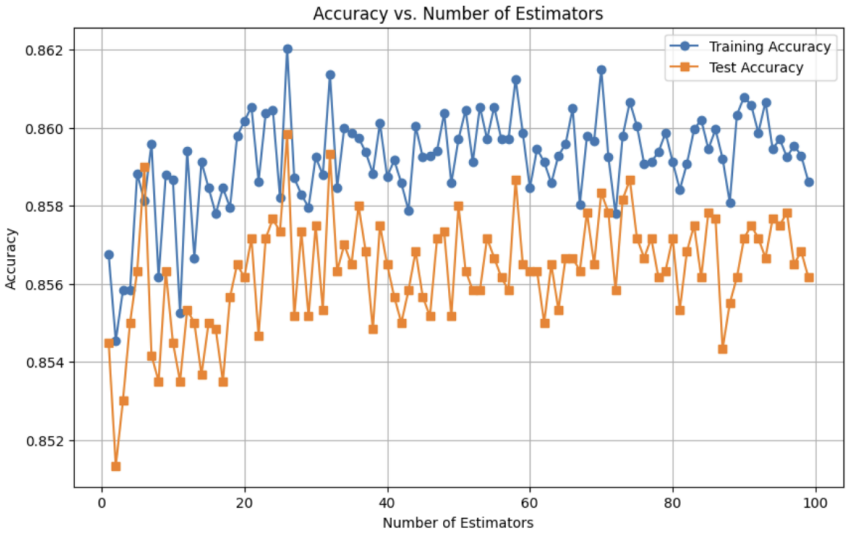
```
model = BaggingClassifierManual()
param_grid = {
    "n_estimators": [10, 15, 25,50,100],
    "max_depth": [5, 7, 10]
}
```

شکل ۲:

: Best Parameters

- **n_estimators :** ۱۰
- **max_depth :** ۵

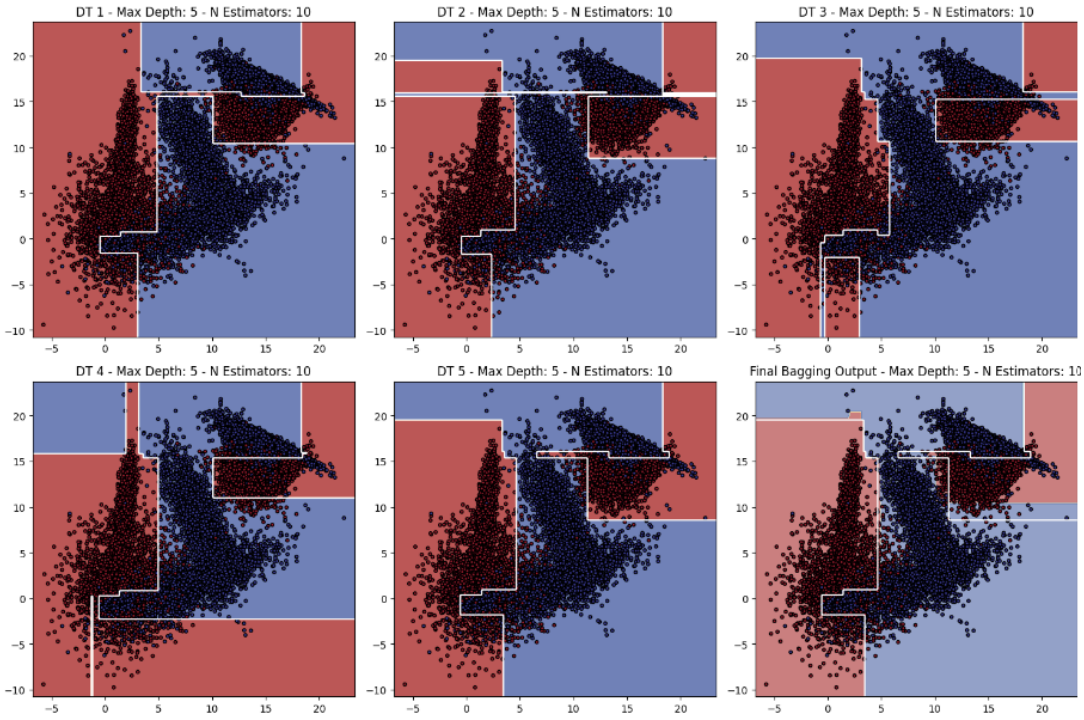
۲.۲) Accuracy by n_estimators



شکل ۳:

با توجه به نمودار بالا که accuracy را فقط با توجه به پارامتر Estimators سنجیده است میتوان به نتیجه رسید که افزایش این پارامتر میتواند تا حدودی مدل را بهتر کند اما صرفا مدل به این پارامتر بستگی ندارد و باید به پارامتر دیگری مثل عمق درخت هم توجه کرد

Decision Bounderies ۳.(۲



شکل ۴:

هر زیرنمودار نشان‌دهنده یکی از مدل‌های پایه (درخت تصمیم) است که با استفاده از داده‌های نمونه‌برداری‌شده‌ی متفاوت (Bootstrap) و با حداکثر عمق (Depth Max) مشخص آموزش دیده است. این درخت‌ها به صورت مستقل از هم آموزش داده می‌شوند.

هر مدل یک مرز تصمیم متفاوت ایجاد کرده است که به طور کامل از دیگر مدل‌ها متمایز است. این تفاوت به دلیل استفاده از زیرمجموعه‌های تصادفی از داده‌ها و ویژگی‌ها است.

آخرین زیرنمودار، نتیجه‌ی نهایی الگوریتم Bagging است. در این نمودار می‌توان مشاهده کرد که مرزهای تصمیم نهایی بسیار هموارتر و دقیق‌تر هستند. ترکیب مدل‌ها باعث شده که خطاهای هر مدل به طور موثری کاهش یافته و عملکرد کلی بهبود پیدا کند.

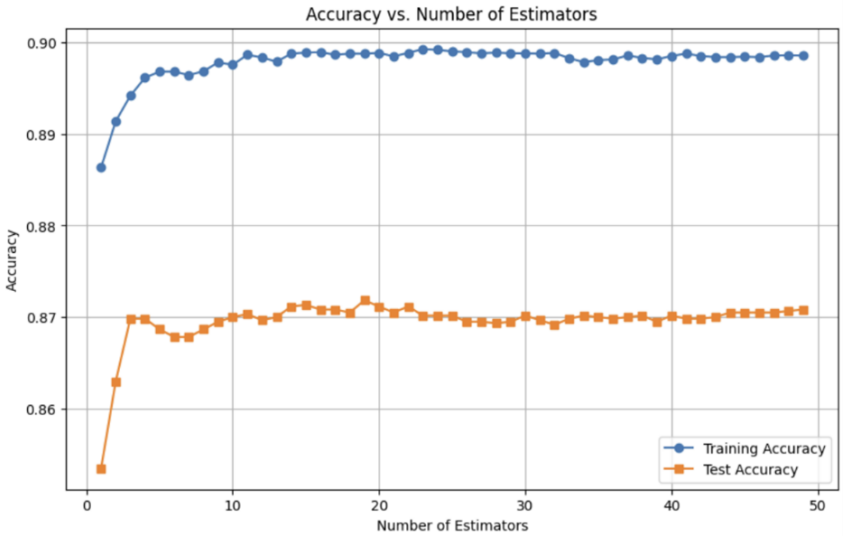
Model Evaluate ۴.(۲

	Training	Test
Accuracy	0.859167	0.858167
Precision	0.859447	0.858291
Recall	0.859167	0.858167
F1 Score	0.859142	0.858146

شکل ۵:

نزدیک بود accuracy در داده های train و test نشان میدهد که مدل خوبی را در خروجی داریم و overfit هم رخ نداده است

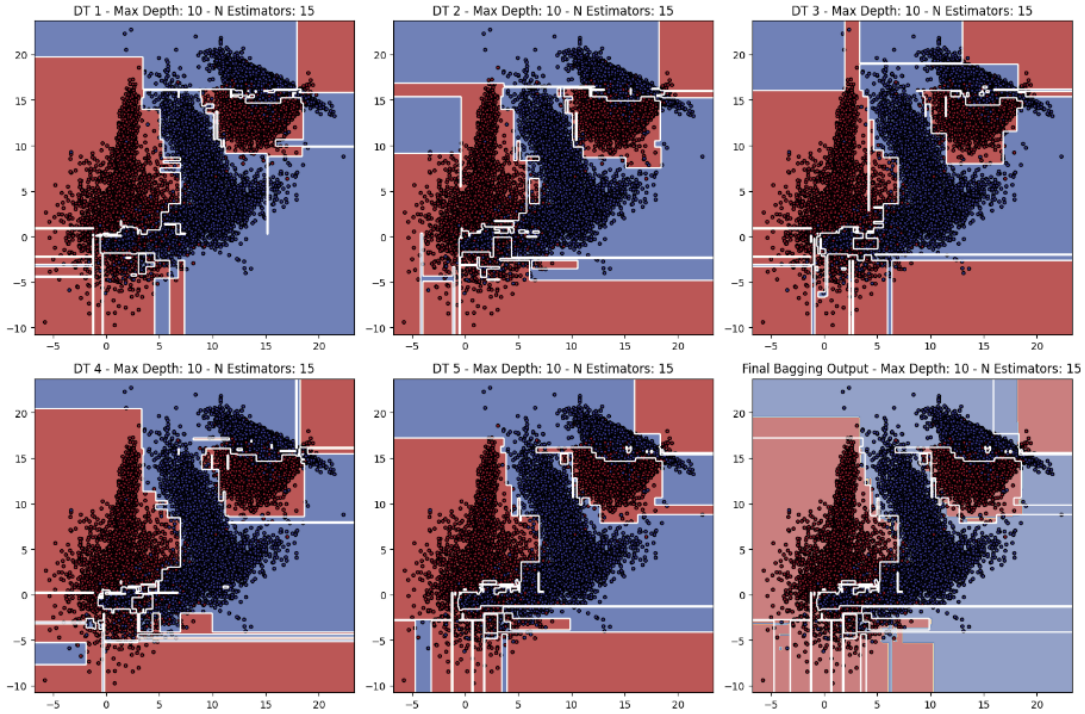
۵.(۲ Accuracy by n_estimators (Library)



شکل ۶:

در این نمودار میبینیم که رشد از تعداد حدودا ۱۵ متوقف شده است و رشد چندان زیادی دیده نمیشود.
یکی از دلایل تفاوت در این نمودار و نمودار قبلی میتواند در این باشد که در اینجا ما از تابع آماده استفاده کرده ایم که میتواند بسیار بهینه تر و بهتر پیاده سازی شده باشد.

Decision Bounderies (Library) ۶.(۲



شکل ۷:

در اینجا هم هر زیرنمودار نشان‌دهنده یکی از مدل‌های پایه (درخت تصمیم) است که با استفاده از داده‌های نمونه‌برداری‌شده‌ی متفاوت (Bootstrap) و با حداکثر عمق (Depth Max) مشخص آموزش دیده است. این درخت‌ها به صورت مستقل از هم آموزش داده می‌شوند. آخرین زیرنمودار، نتیجه‌ی نهایی الگوریتم Bagging است. چندان تفاوتی با کلاس بندی تابع دستی ندارد

Model Evaluate (Library) ۷.(۲

	Training	Test
Accuracy	0.898875	0.871333
Precision	0.899269	0.871802
Recall	0.898875	0.871333
F1 Score	0.898848	0.871306

شکل ۸:

این ارزیابی نشان میدهد که تفاوت چندان زیادی بین تابعی که به صورت دستی پیاده سازی شده و تابع آماده در کتابخانه sklearn نیست و نتیجه خوبی به ما داده است

RandomForest (۳

Training ۱.(۳

: GridSearch Parameters

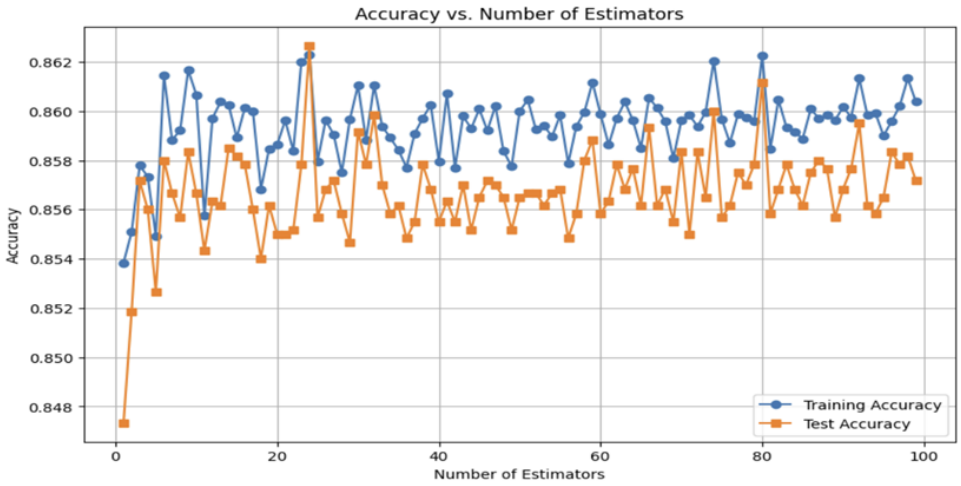
```
model = RandomForestManual()
param_grid = {
    "n_estimators": [10, 15],
    "max_depth": [5, 7, None],
    "max_features": [None,'sqrt', 'log2']
}
```

شکل ۹:

: Best Parameters

- **n_estimators :** ۱۰
- **max_depth :** ۱۵
- **max_features :** None

Accuracy by n_estimators ۲.(۳

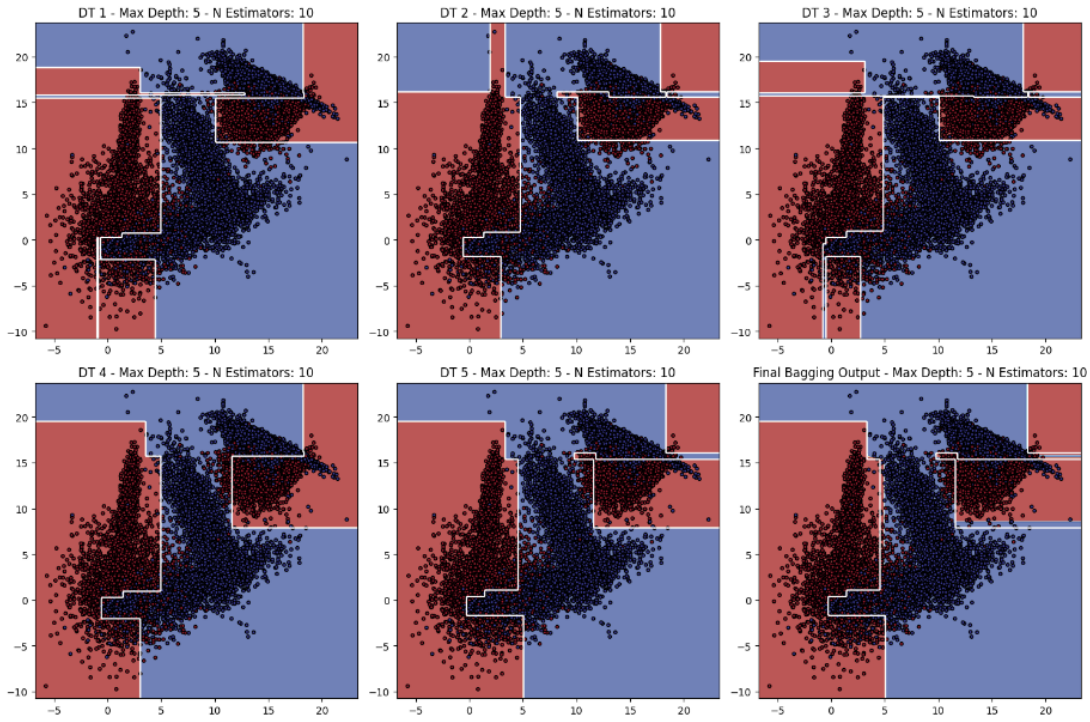


شکل ۱۰:

با توجه به نمودار بالا که accuracy را فقط با توجه به پارامتر Estimators سنجیده است میتوان به نتیجه رسید که افزایش این پارامتر میتواند تا حدودی مدل را بهتر کند اما صرفا مدل به این پارامتر بستگی ندارد و باید به پارامتر دیگری مثل عمق درخت هم توجه کرد

در این نمودار دیده میشود که در یکی از مقادیر accuracy داده تست بیشتر شده است این به دلیل بهینه نبودن تابع میتواند باشد که منجر به این اتفاق شده است

۳.(۳ Decision Bounderies



شکل ۱۱:

در ۵ نمودار اول هر درخت تصمیم‌گیری (DT) تنها بخشی از داده‌ها را از طریق Sampling Bootstrap دریافت می‌کند و بر اساس زیرمجموعه‌ای از ویژگی‌ها آموزش می‌بیند. مرزهای تصمیم درخت‌ها به صورت تکه تکه هستند، زیرا:

- هر درخت فقط بر اساس داده‌های محدود آموزش داده شده است.
- محدودیت در max_depth باعث می‌شود مدل بیش از حد پیچیده نشود و مرزهای ساده‌تری ارائه کند.
- این مرزها ممکن است به تنهایی دقیق نباشند و منجر به پیش‌بینی‌های نادرست شوند.

نمودار آخر (پایین سمت راست) نشان‌دهنده خروجی نهایی الگوریتم Random Forest است. این خروجی با استفاده از رأی‌گیری اکثریت (majority voting) بین پیش‌بینی‌های تمام درخت‌ها به دست آمده است.

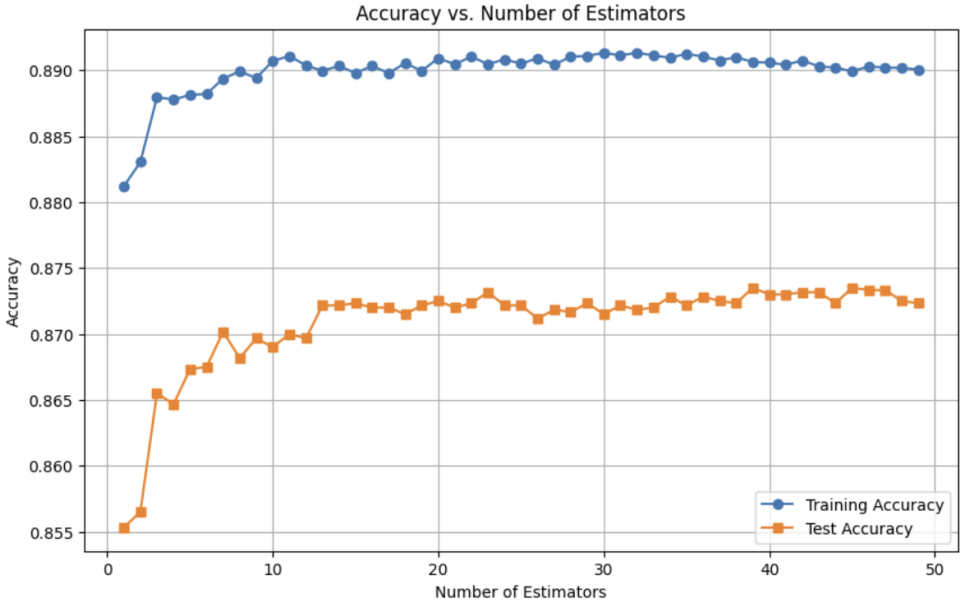
۴.(۳ Model Evaluate

	Training	Test
Accuracy	0.857000	0.854000
Precision	0.857324	0.854176
Recall	0.857000	0.854000
F1 Score	0.856971	0.853972

شکل ۱۲:

ارزیابی این مدل با پارامتر های انتخاب شده به شرح زیر است که برای دیتا درصد نسبتا خوبی را داده است و overfit هم رخ نداده است.

۵.(۳ Accuracy by n_estimators (Library)

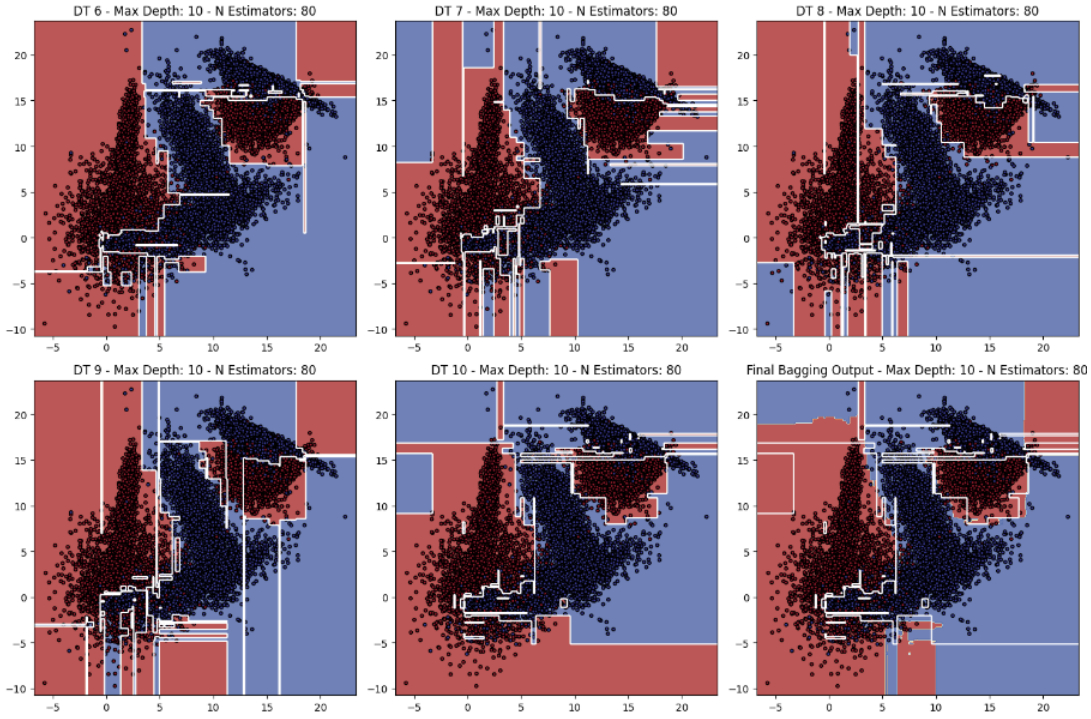


شکل ۱۳:

در این نمودار میبینیم که رشد از تعداد حدودا ۱۳ متوقف شده است و رشد چندان زیادی دیده نمیشود و حتی در مقدار های ۴۰ به بعد اندکی کاهش پیدا کرده است.

یکی از دلایل تفاوت در این نمودار و نمودار قبلی میتواند در این باشد که در اینجا ما از تابع آماده استفاده کرده ایم که میتواند بسیار بهینه تر و بهتر پیاده سازی شده باشد.

۶.(۳ Decision Bounderies (Library)



شکل ۱۴:

این درخت‌ها به دلیل تکنیک‌های زیرمجموعه سازی ویژگی‌ها و داده‌ها که در جنگل تصادفی استفاده می‌شوند، پیش‌بینی‌های متفاوتی ارائه می‌دهند.

اختلاف در این گراف‌ها نشان‌دهنده این است که هر درخت به صورت جزئی داده‌ها را مدل می‌کند.

در گراف‌های DT۱ تا DT۵، مرزهای تصمیم اغلب ناپایدار و غیر دقیق هستند. این رفتار به دلیل پیچیدگی محدود هر درخت تصمیم و عدم تعمیم‌پذیری مناسب یک درخت است.در گراف نهایی، مرزهای تصمیم بهبود یافته و صاف‌تر شده‌اند، زیرا ترکیب درخت‌ها نویز را کاهش داده و منجر به مدل قوی‌تری شده است.

۷.(۳ Model Evaluate (Library)

	Metric	Train	Test
0	Accuracy	0.894542	0.871500
1	Precision	0.895000	0.872111
2	Recall	0.894542	0.871500
3	F1 Score	0.894508	0.871462

شکل ۱۵:

حدود دو درصد در تست و ۴ درصد در تمرین بهبود داشته است و در کل دقت قابل قبول و تعمیم پذیری بالایی را نتیجه می‌دهد.

AdaBoost (۴

Training ۱.(۴

: GridSearch Parameters

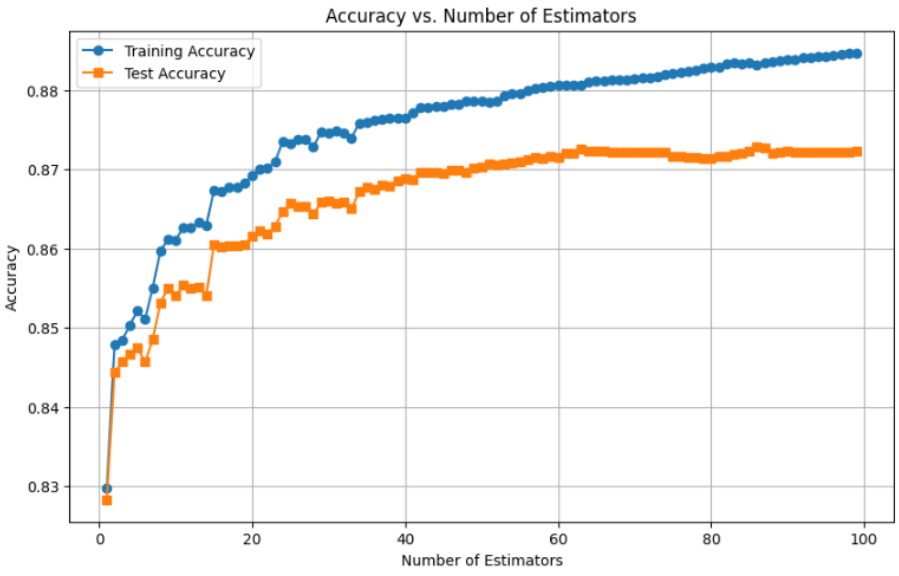
```
param_grid = {
    "n_estimators": [50, 100, 200],
    "learning_rate": [0.01, 0.1, 1],
    'estimator': [DecisionTreeClassifier(max_depth=1),
                  DecisionTreeClassifier(max_depth=2),
                  DecisionTreeClassifier(max_depth=3),
                  DecisionTreeClassifier(max_depth=4),
                  DecisionTreeClassifier(max_depth=5)]
}
```

شکل ۱۶:

: Best Parameters

- **n_estimators :** ۱۰۰
- **learning_rate :** ۰.۱
- **estimator:** DecisionTree with max_depth = ۴

Accuracy by n_estimators ۲.(۴



شکل ۱۷:

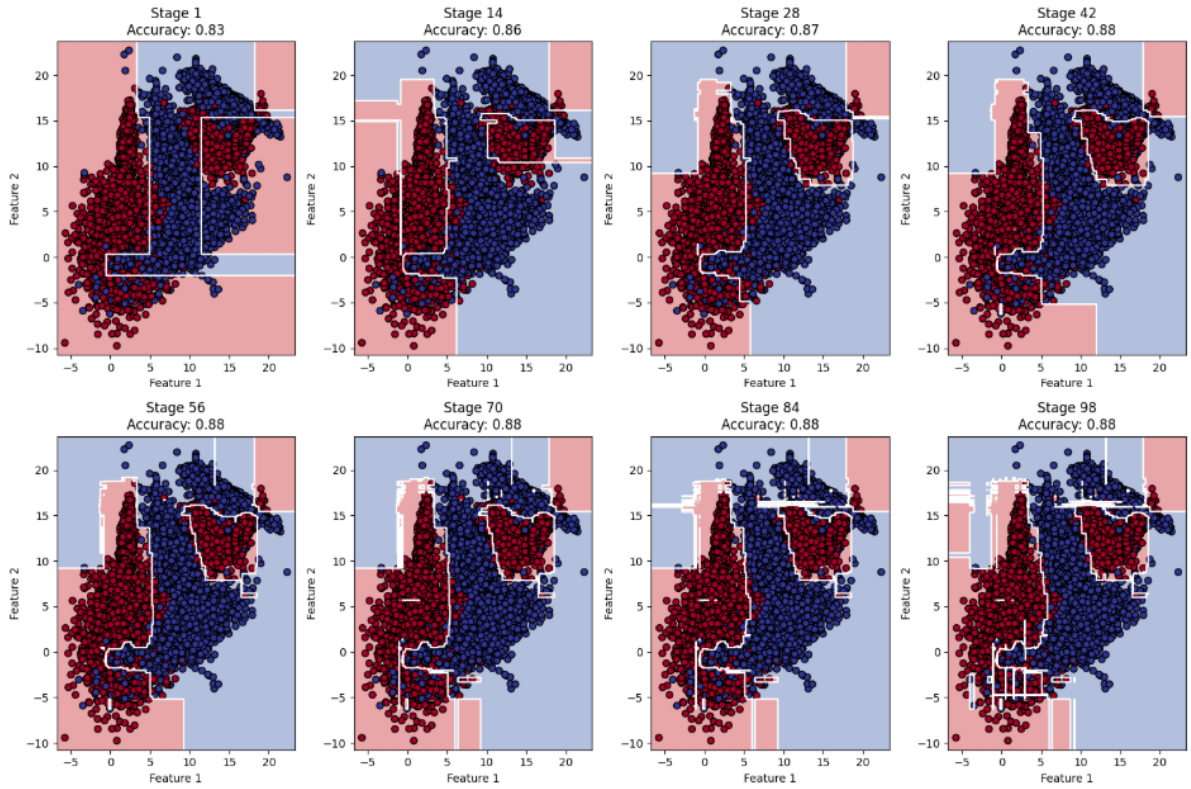
: Training Data

با افزایش تعداد تخمین زن ها، خطای داده‌های آموزشی را به صورت تکراری کاهش دهد. الگوریتم به تطبیق با داده‌های آموزشی ادامه می‌دهد و باعث افزایش دقت روی مجموعه آموزشی می‌شود. اما شیب افزایش دقت به مرور کاهش می‌یابد چون الگوریتم با داده های آموشی تطبیق پیدا می‌کند.

: Test Data

دقت داده‌های تست در ابتدا افزایش می‌یابد، زیرا مدل با تعداد تخمین‌زن‌های بیشتر بهتر تعمیم می‌دهد. اما پس از یک نقطه مشخص(حدود ۶۰ تخمین زن)، شیب تقریباً صفر می‌شود، که نشان می‌دهد اضافه کردن تخمین‌زن‌های بیشتر تأثیر قابل توجهی در بهبود تعمیم ندارد. این مسطح شدن به این دلیل اتفاق می‌افتد که AdaBoost ممکن است با افزایش بیش از حد تعداد تخمین‌زن‌ها، بیش از حد به داده‌های آموزشی تطبیق پیدا کند و پیچیدگی اضافی دیگر به پیش‌بینی بهتر داده‌های جدید کمکی نمی‌کند.

Decision Bounderies ۳.(۴



شکل ۱۸:

در این قسمت میتوانیم مرز بین طبقه‌بندی هارا مشاهده کنیم. همانطور که در تصاویر مشخص است، هرچقدر به پایان الگوریتم نزدیک تر می‌شویم دقت افزایش می‌یابد و از دقت ۸۳ درصد به ۸۷ درصد رسیده‌ایم.

همچنین در بهبود مرزهای تصمیم گیری به عنوان مثال بین مرحله ۱ و ۱۴ ،داده های سمت راست کلاس آبی، بخش زیادی از آنها در ابتدا به کلاس قرمز پیشبینی می‌شده اند که در مرحله ۱۴ آبی پیشبینی شده‌اند. اما با رفع این مشکل بخشی از داده‌های قرمز(بین ۱۰ و ۲۰ افقی و ۱۰ و ۸ عمودی)به کلاس آبی پیش‌بینی شده اند که در مرحله ۲۸ این مشکل رفع شده.

هرچقدر مرحله جلوتر می‌رود مدل داده های جزئی که ممکن است از تراکم کلاس خودشان دور باشه اند هم درست پیشبینی میکند مثلا دو داده کلاس آبی در مست چپ(بین -۵ و۰ افقی و -۸ و -۲ عمودی) که تا قبل مرحله ۹۸ برای کلاس قرمز پیشبینی می‌شده اند نیز به درستی پیش‌بینی خواهند شد. حتی نواحی که در آنها داده ای نیست نیز تقسیم بندی شده اند(بین ۵ و ۱۰ افقی و -۱۰ و -۵ عمودی) یا بخش بزرگی از دیتا (بین -۱۰ و -۵ عمودی) در ابتدا کاملا کلاس قرمز پیشبینی می‌شد و به مرور حدودا این ناحیه نصف برای کلاس آبی شد و نصف برای کلاس قرمز.

در کل میتواند دید هرچه جلوتر می‌رویم نواحی کلاس ها از مربعی بودن تغییر میکند و شکل دیتا را به خود میگیرد مخصوصا برای کلاس قرمز(بین ۱۰ و ۲۰ افقی و ۱۰ و ۱۵ عمودی) که کاملا شکل کلاس قرمز را به خود گرفته.

۴.(۴ Model Evaluate

	Training	Test
Accuracy	0.884708	0.872333
Precision	0.884827	0.872589
Recall	0.884708	0.872333
F1 Score	0.884698	0.872321

شکل ۱۹:

تفاوت بین دقت داده های تمرین و تست کمتر از ۲ درصد است که می‌توان نتیجه گرفت مدل اوورفیت نشده و داده های تمرین را حفظ نکرده است و دارای تعمیم پدیری خیلی خوبی است. همچنین معیار های ارزیابی تقریبا مثل هم هستند که این موضوع نشان می‌دهد مدل توانایی بالایی در تشخیص صحیح کلاس‌های مثبت و منفی دارد، بدون این‌که به نفع یک کلاس خاص متمایل شود.

۵) Stacked Learners

۵.۱) Training

: GridSearch Parasmeters and Classifires

```
classifiers_and_params = [
    (RandomForestClassifier(), {'n_estimators': [50, 100], 'max_depth': [None,5, 10],
                                'min_samples_split': [5, 10],'min_samples_leaf': [1, 2,5]}),
    (SVC(probability=True), {'C': [0.1, 1, 10], 'kernel': ['rbf'],'gamma':['auto','scale']}),
    (LogisticRegression(), {'C': [0.1, 1, 10]}),
    (KNeighborsClassifier(), {'n_neighbors': [3, 5, 7]})
]
```

شکل ۲۰:

: Best Parameters

- **RandomForest :** n_estimators: ۱۰۰، min_samples_split: ۵، min_samples_leaf: ۱ ، max_depth: ۱۰
- **SVM :** kernel: rbf ، gamma: auto ، C: ۱۰
- **LogisticRegression :** C: ۰.۱
- **KNN :** n_neighbors: ۷

۵.۲) Model Evaluate

	Training	Test
Accuracy	0.895083	0.862000
Precision	0.895089	0.862060
Recall	0.895083	0.862000
F1 Score	0.895083	0.861999

شکل ۲۱:

با توجه به معیارهای ارزیابی میتوان دید کلسیفایر نهایی ما دارای دقت خوبی است و همچنین اختلاف بین معیار ها در داده تست و تمرین حدود ۳ درصد است که میتوان تعمیم پذیری خوبی را نتیجه گرفت. با توجه به اینکه کلسیفایر ها بسیار متنوع هستند، برای دستیابی به بالاترین درصد ممکن میتوان کلسیفایر های بیشتر برای Base Learners و کلسیفایر های مختلف را به عنوان کلسیفایر نهایی امتحان کرد.

(Extra) overfitting ۳.(۵

	Training	Test
Accuracy	1.0	0.819333
Precision	1.0	0.819344
Recall	1.0	0.819333
F1 Score	1.0	0.819328

شکل ۲۲: قبل جلوگیری از اوورفیت

	Training	Test
Accuracy	0.912083	0.866667
Precision	0.912222	0.866945
Recall	0.912083	0.866667
F1 Score	0.912075	0.866652

شکل ۲۳: بعد از جلوگیری از اورفیت

در این قسمت یک درخت تصمیم اوورفیت شده به کلسیفایر های قبلی اضافه کردیم. همانطور که در در عکس ها مشخص هست فقط با اووفیت شدن یک مدل، مدل نهایی اوورفیت شده است و فاصله دقت بین داده آموزش و تست حدود ۱۹ درصد است. با اجرای الگوریتم برای جلوگیری از اوورفیت فاصله دقت داده های آموزش و تست به ۵ درصد کاهش یافت و دقت روی داده‌های تست مثل قبل شد که نشان می‌دهد ما با موفقیت از اوورفیت شدن مدل جلوگیری کرده ایم.