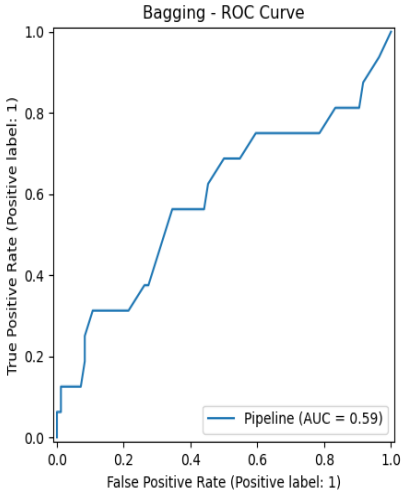
Customer_chrn.scv

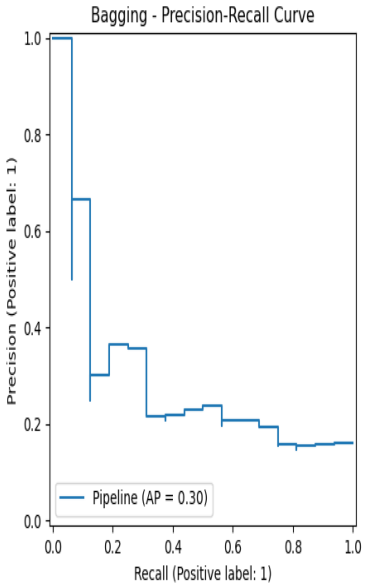
0. model

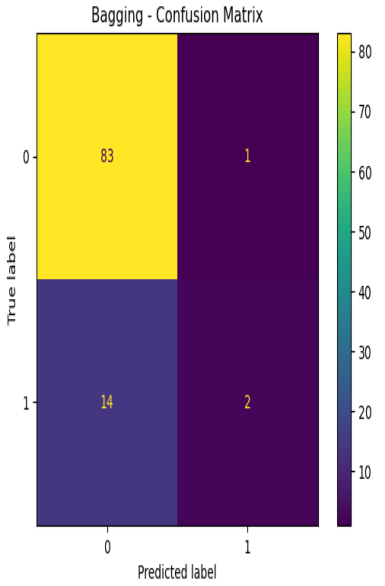
1. Bagging Classfier

2. Random Forest(RandomForestClassfier)

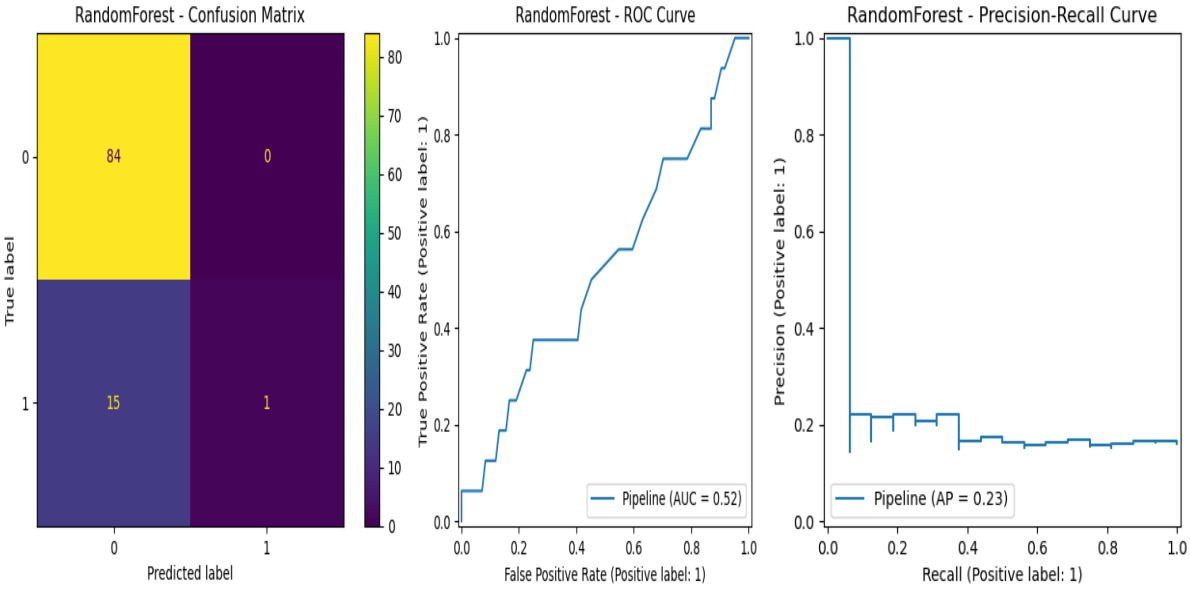
3. AdaBoost



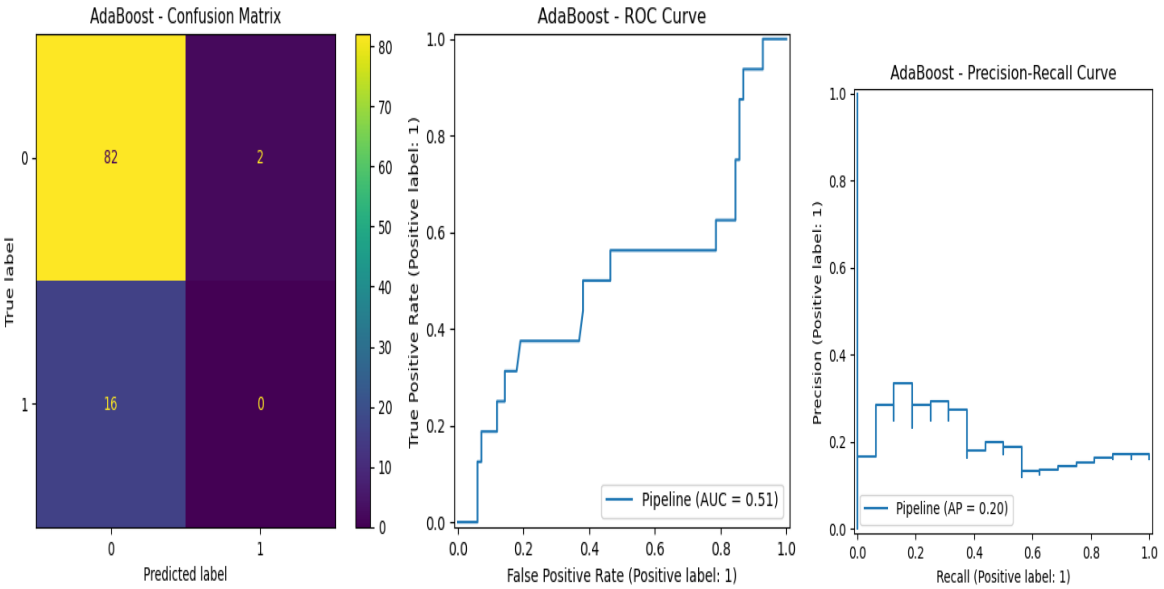





```
K-Fold CV Accuracy: 0.8220 (std: 0.0240)

Accuracy: 0.8500 | Precision: 0.6667 | Recall: 0.1250
              precision    recall  f1-score   support

           0       0.86      0.99      0.92        84
           1       0.67      0.12      0.21        16

    accuracy                           0.85       100
   macro avg       0.76      0.56      0.56       100
weighted avg       0.83      0.85      0.80       100
```




```
=== RandomForest ===
K-Fold CV Accuracy: 0.8260 (std: 0.0162)
Accuracy: 0.8500 | Precision: 1.0000 | Recall: 0.0625
              precision    recall  f1-score   support

           0       0.85      1.00      0.92        84
           1       1.00      0.06      0.12        16

    accuracy                           0.85       100
   macro avg       0.92      0.53      0.52       100
weighted avg       0.87      0.85      0.79       100
```




```
K-Fold CV Accuracy: 0.8260 (std: 0.0102)
```

```
Accuracy: 0.8200 | Precision: 0.0000 | Recall: 0.0000
              precision    recall  f1-score   support

           0       0.84      0.98      0.90        84
           1       0.00      0.00      0.00        16

    accuracy                           0.82       100
   macro avg       0.42      0.49      0.45       100
weighted avg       0.70      0.82      0.76       100
```

1. 평균 정확도 :

K-Fold Cv Accuracy 정확도는 0.8260으로 RandeomForestClassifier 모델과 AdaBoost 모델이 같게 나왔다.

Bagging Classfier은 0.8260으로 평균 정확도가 나왔다.

AUC :

정확도는

1. Bagging Classfier : 0.59

2. Random Forest 0.52

3. AdaBoost 0.51

로 나왔다. (Roc curve 그래프참고)

학습시간도 계산 :

1. Bagging Classfier : `0.3943 seconds`

```
Training Time: 0.3943 seconds
```

2. Random Forest(RandomForestClassfier) : `0.2434 seconds`

```
Training Time: 0.2434 seconds
```

3. AdaBoost : `0.2260 seconds`

```
Training Time: 0.2260 seconds
```

---

2. 3개의 모셀 성능 표로 정리

Bagging

	precision	Recall	F1-socre	Support
0	0.86	0.99	0.92	84
1	0.67	0.12	0.221	16
Accuracy			0.85	100
Macro avg	0.76	0.56	0.56	100
Weighted avg	0.83	0.85	0.80	100

Random Forest

	precision	Recall	F1-socre	Support
0	0.85	1.00	0.92	84
1	1.00	0.06	0.12	16
Accuracy			0.85	100
Macro avg	0.92	0.53	0.52	100
Weighted avg	0.87	0.85	0.79	100

AdaBoost

	precision	Recall	F1-socre	Support
0	0.84	0.98	0.90	84
1	0.00	0.00	0.00	16
Accuracy			0.85	100
Macro avg	0.42	0.49	0.45	100
Weighted avg	0.70	0.82	0.76	100

비교해서 분석한 결과 Accuracy : 0.8500

Precision1.0000

Recall0.0625

그리고 Confusion Matrix에서 Tp와 Tn의 값이 다른 모델보다 값이 좋다.

그렇기에 제일 우수한 모델은 RandomForest(RandomForestClassfier) 이다.

3. 모델별 과적합 가능성, 학습속도, 데이터 특성과의 궁합

모델별 과적합 가능성

Train. Accuracy: 1.0000

Test Accuracy : 0.8500

훈련데이터에 너무 과도하게 맞춰져서 높은 성능을 보이지만 test에서는 떨어지는것을 보아하니 과적합 가능성이 있다.

학습 속도

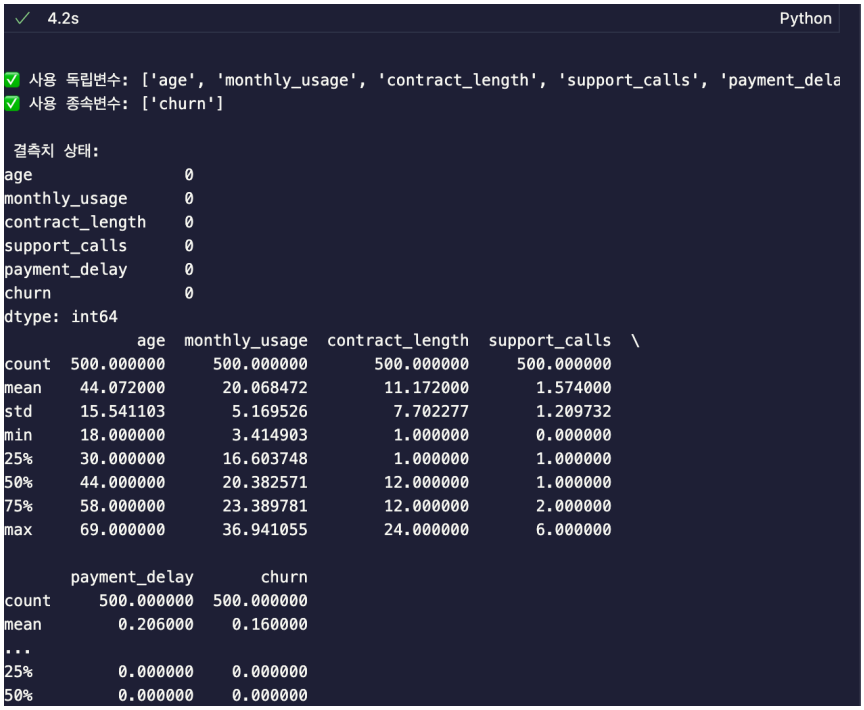
모델을 각 주석처리후 세션을 재시작한후 학습 속도에 대해 비교해봤다.

여러 개의 모델을 동시에 학습시켜야 하므로 시간과 계산 비용 조금 더 들어서 제일 느렸다고 생각한다.

1. Bagging Classfier 5.3s

2. Random Forest(RandomForestClassfier) 4.4s

3. AdaBoost



1. Bagging Classfier

```
=== Bagging ===
Training Time: 0.3943 seconds
```

2. Random Forest(RandomForestClassfier) 4.4s

```
=== RandomForest ===
Training Time: 0.2434 seconds
```

3. AdaBoost

```
=== AdaBoost ===
Training Time: 0.2260 seconds
```

위에 결과대로 AdaBoost > Random Forest > Baggig Classfier 순으로 속도가 나왔다.

데이터 특성과의 궁합

Chrn(이탈 여부)를 타겟으로 하는 분류 모델을 만들었는데

Age 고객 나이 (연령에 따른 사용 패턴 차이)

monthly_usage(월 평균 사용량)

contract_length 계약 기간- (계약 기간과 이탈 위험 연관성 높을 수 있음(장기 계약 고객은 이탈 적음)

support_calls (문의 횟수 많으면 불만족 표시 또는 문제 발생 신호)

payment_delay - (지불 지연은 이탈 위험 신호)

라고 중요하다고 생각해 독립변수로 두고 모델을 돌렸다.