타이타닉 승객 정보를 이용한 생존자 예측

4조

김동규, 김채윤, 문희원, 이주환, 이중기

목차

1. 목적

- 데이터를 통해 생존자의 생사여부와 다른 데이터들 간의 연관성을 분석하여 생존에 영향을 미치는 요소를 찾아내는 것.

2. 데이터 프로파일링 리포트

Overview

Dataset info

Number of variables	12
Number of observations	891
Missing cells	866 (8.1%)
Duplicate rows	0 (0.0%)
Total size in memory	83.6 KiB
Average record size in memory	96.1 B

Variables types

Numeric	5
Categorical	5
Boolean	1
Date	0
URL	0
Text (Unique)	1
Rejected	0
Unsupported	0

Warnings

`Age` has 177 (19.9%) missing values	Missing
`Cabin` has a high cardinality: 148 distinct values	Warning
`Cabin` has 687 (77.1%) missing values	Missing
`Fare` has 15 (1.7%) zeros	Zeros
`Parch` has 678 (76.1%) zeros	Zeros
`SibSp` has 608 (68.2%) zeros	Zeros
`Ticket` has a high cardinality: 681 distinct values	Warning

3. 데이터 전처리

- 결측치 채우기
- 불필요한 열 제거하기
- 텍스트로 되어있는 요소는 숫자로 바꿔주기
- 실수 범위를 구간 범위로 바꿔주기
- 정규화

4. 데이터 전처리

- 결측치 채우기

```
1  x_data['Age'].isnull().sum()
```

```
177
```

Age 요소에
비어 있는 값이 존재

```
mean_age = x_data['Age'].median(skipna=True)
x_data['Age'] = x_data['Age'].fillna(mean_age)
```

```
1  x_data['Age'].isnull().sum()
```

```
0
```

4. 데이터 전처리

- 텍스트로 되어있는 요소는 숫자로 바꿔주기

```
x_data['Sex'] = x_data['Sex'].apply(lambda x: 0 if x == 'male' else 1 )
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	C85	C
2	3	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	113803	53.1000	C123	S
4	5	3	Allen, Mr. William Henry	0	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
886	887	2	Montvila, Rev. Juozas	0	27.0	0	0	211536	13.0000	NaN	S
887	888	1	Graham, Miss. Margaret Edith	1	19.0	0	0	112053	30.0000	B42	S
888	889	3	Johnston, Miss. Catherine Helen "Carrie"	1	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	Behr, Mr. Karl Howell	0	26.0	0	0	111369	30.0000	C148	C
890	891	3	Dooley, Mr. Patrick	0	32.0	0	0	370376	7.7500	NaN	Q

4. 데이터 전처리

- 실수 범위를 구간 범위로 바꿔주기

```
x_data.loc[ x_data['Ag
x_data.loc[(x_data['Ag
x_data.loc[(x_data['Ag
x_data.loc[(x_data['Ag
x_data.loc[(x_data['Ag
x_data.loc[(x_data['Ag
x_data.loc[(x_data['Ag
x_data.loc[(x_data['Ag
x_data.loc[(x_data['Ag
x_data.loc[ x_data['Ag
```

	Pclass	Sex	Age	SibSp	Parch	Age_group
0	3	0	22.0	1	0	3.0
1	1	1	38.0	1	0	6.0
2	3	1	26.0	0	0	3.0
3	1	1	35.0	1	0	5.0
4	3	0	35.0	0	0	5.0
...	...	...	...	...	...	...
886	2	0	27.0	0	0	4.0
887	1	1	19.0	0	0	2.0
888	3	1	28.0	1	2	4.0
889	1	0	26.0	0	0	3.0
890	3	0	32.0	0	0	5.0

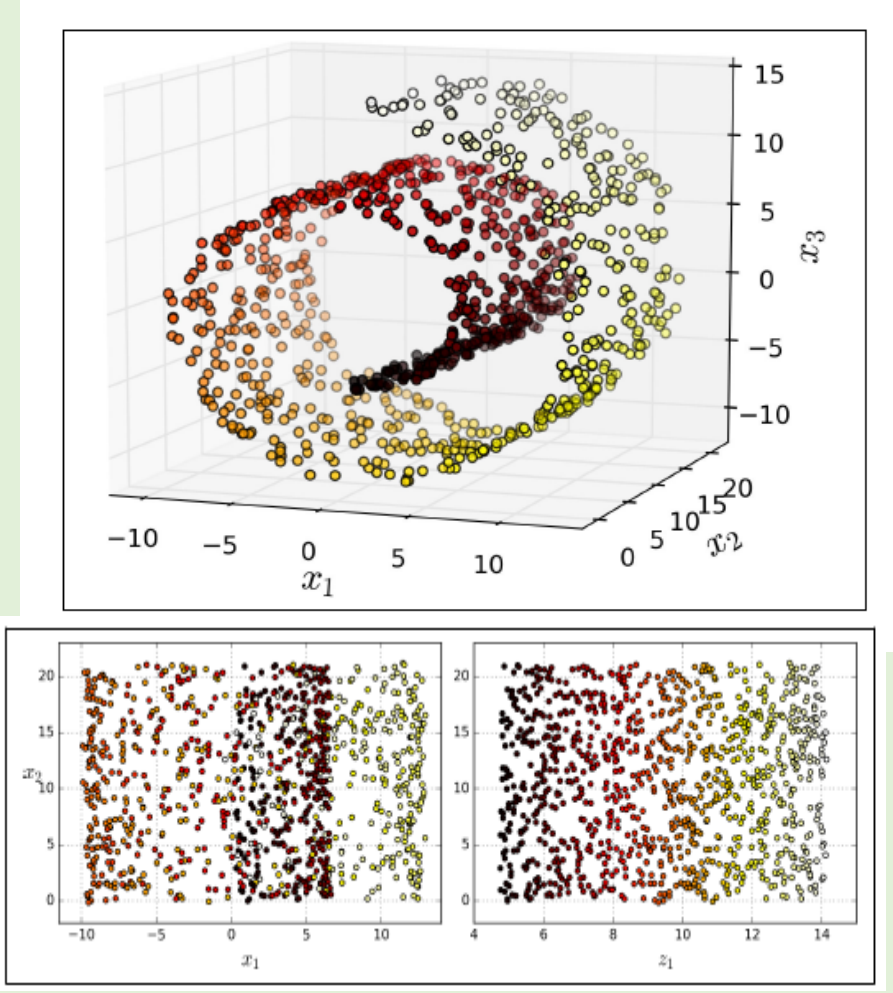
```
16), 'Age_group'] = 1
20), 'Age_group'] = 2
26), 'Age_group'] = 3
30), 'Age_group'] = 4
36), 'Age_group'] = 5
40), 'Age_group'] = 6
46), 'Age_group'] = 7
50), 'Age_group'] = 8
60), 'Age_group'] = 9
```

4. 데이터 전처리

- 불필요한 열 제거하기

	PassengerId	Pclass			Pclass	Sex	Age	SibSp	Parch		Parch	Ticket	Fare	Cabin	Embarked
0	1	3		0	3	0	22.0	1	0		0	A/5 21171	7.2500	NaN	S
1	2	1	Cumings, Mrs. John	1	1	1	38.0	1	0		0	PC 17599	71.2833	C85	C
2	3	3		2	3	1	26.0	0	0		0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	Futrelle, Mrs.	3	1	1	35.0	1	0		0	113803	53.1000	C123	S
4	5	3		4	3	0	35.0	0	0		0	373450	8.0500	NaN	S
...	...	...		...	...	...	...	...	...		...	...	...	...	...
886	887	2		886	2	0	27.0	0	0		0	211536	13.0000	NaN	S
887	888	1		887	1	1	19.0	0	0		0	112053	30.0000	B42	S
888	889	3	Johnston,	888	3	1	28.0	1	2		2	W./C. 6607	23.4500	NaN	S
889	890	1		889	1	0	26.0	0	0		0	111369	30.0000	C148	C
890	891	3		890	3	0	32.0	0	0		0	370376	7.7500	NaN	Q

4. 주성분 분석

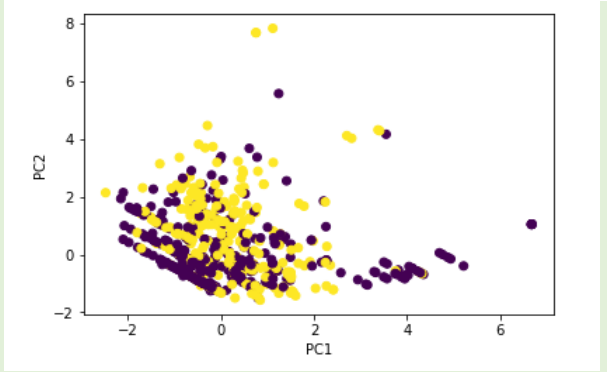


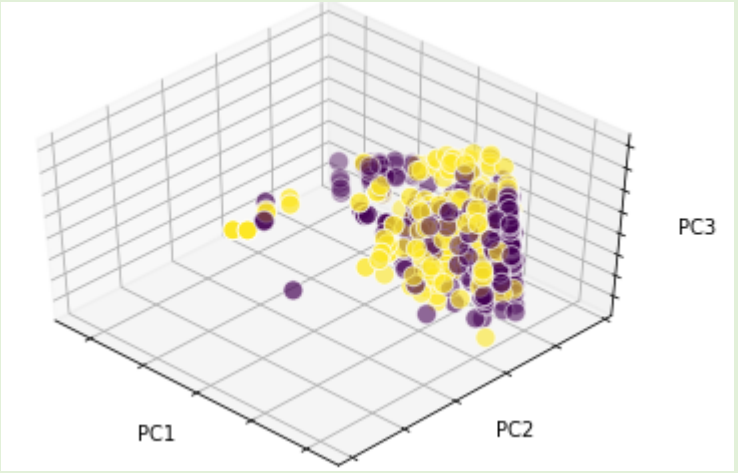
- 주성분 분석은 데이터의 분산을 최대한 보존하면서 서로 직교하는 새 기저를 찾아, 고차원 공간의 표본들을 선형 연관성이 없는 저차원 공간으로 변환하는 기법

```
modelp = decomposition.PCA(n_components=3)
```

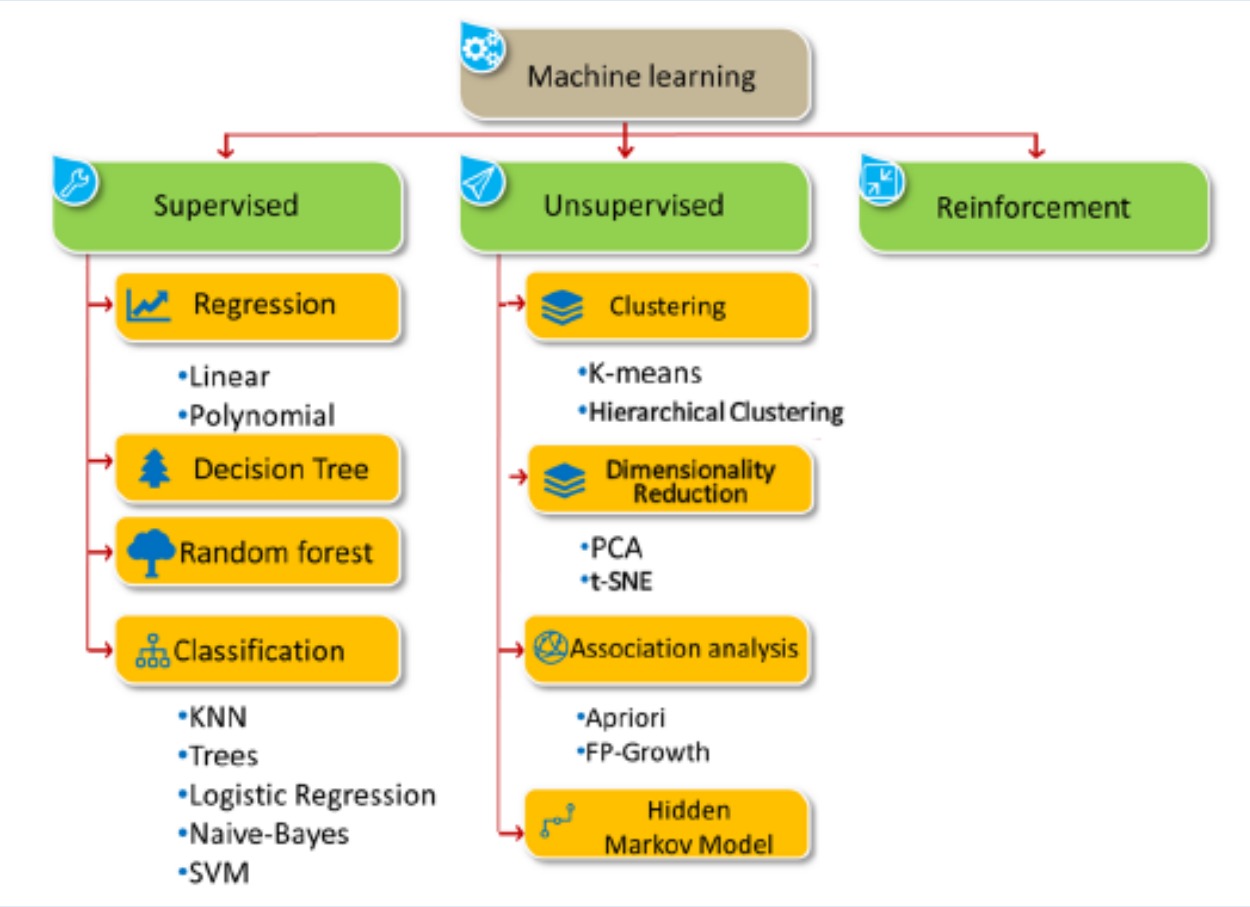
```
print(modelp.explained_variance_ratio_)
print(np.sum(modelp.explained_variance_ratio_))
```

```
[0.35181484 0.32182483 0.15948981]
0.8331294777381699
```



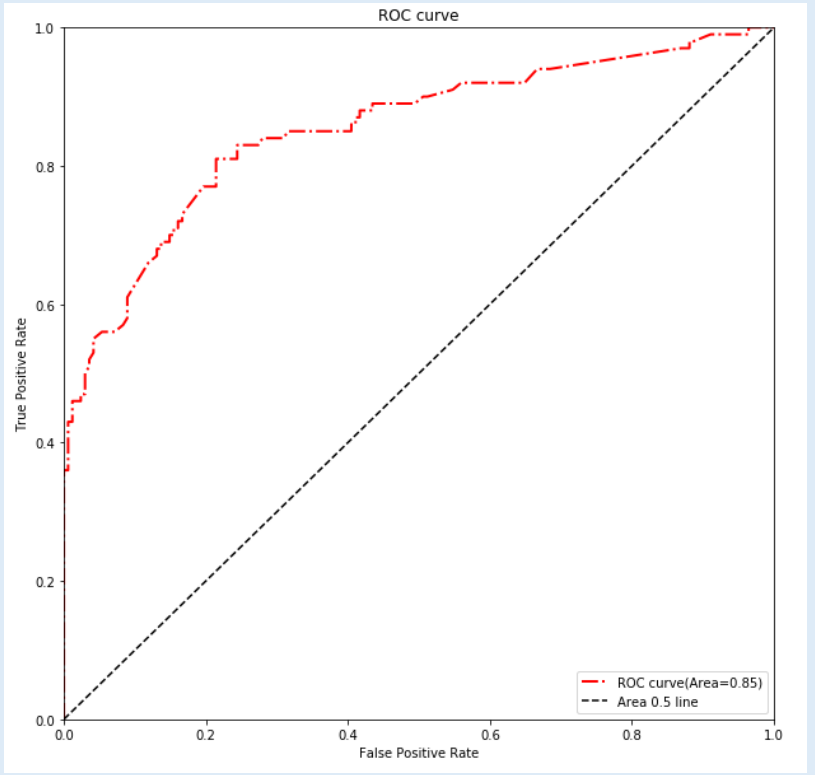


5. 머신러닝 모델


Machine learning
Supervised
Unsupervised
Reinforcement
Regression
•Linear
•Polynomial
Decision Tree
Random forest
Classification
•KNN
•Trees
•Logistic Regression
•Naive-Bayes
•SVM
Clustering
•K-means
•Hierarchical Clustering
Dimensionality Reduction
•PCA
•t-SNE
Association analysis
•Apriori
•FP-Growth
Hidden Markov Model

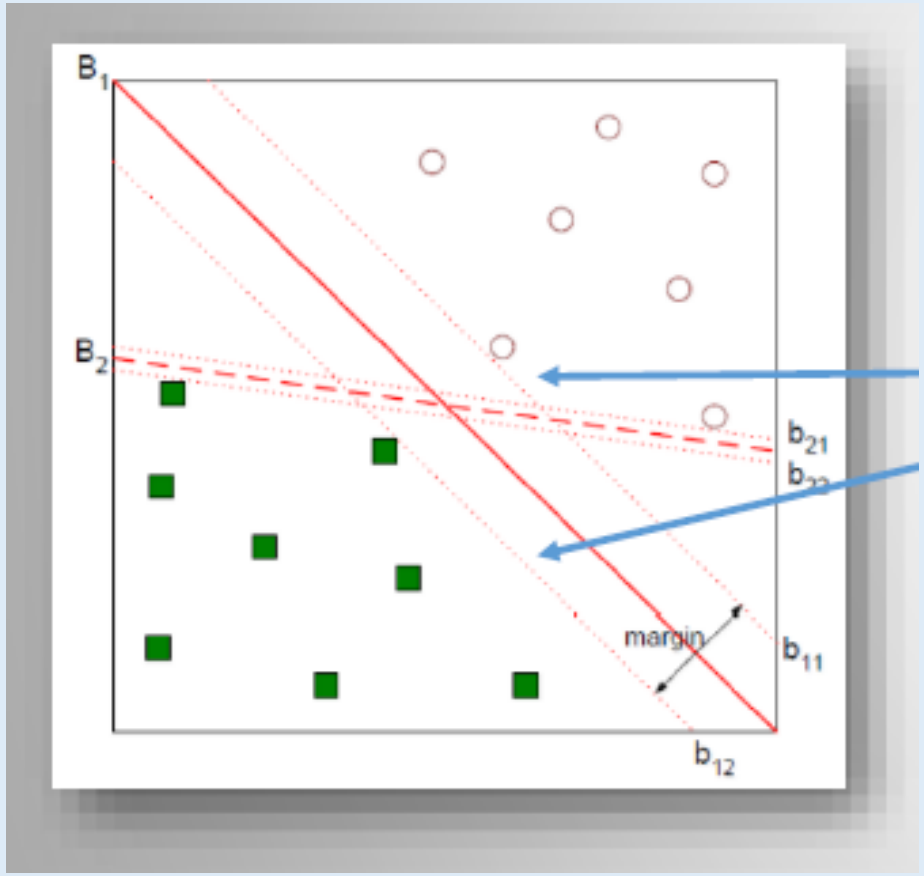
5. 로지스틱 회귀(Logistic Regression)

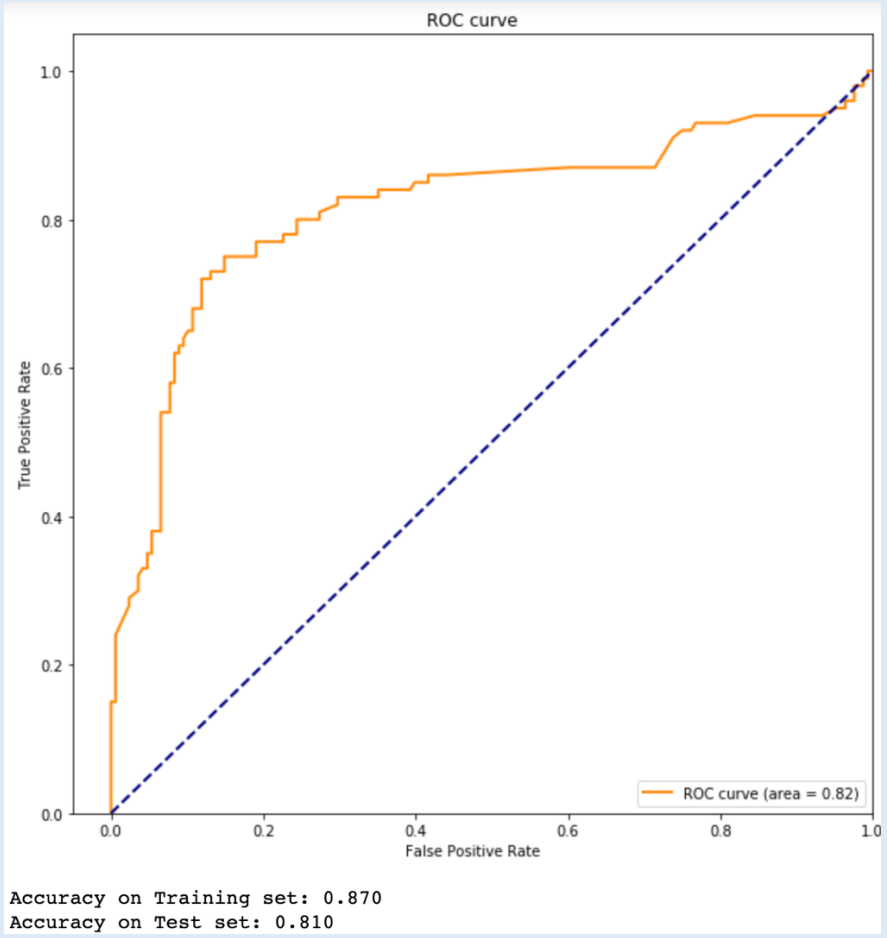

```
Train_Accuracy:  0.797752808988764
Test_Accuracy:  0.7910447761194029
```



AUC 0.85

5. 서포트 벡터 머신(SVM)





AUC 0.82

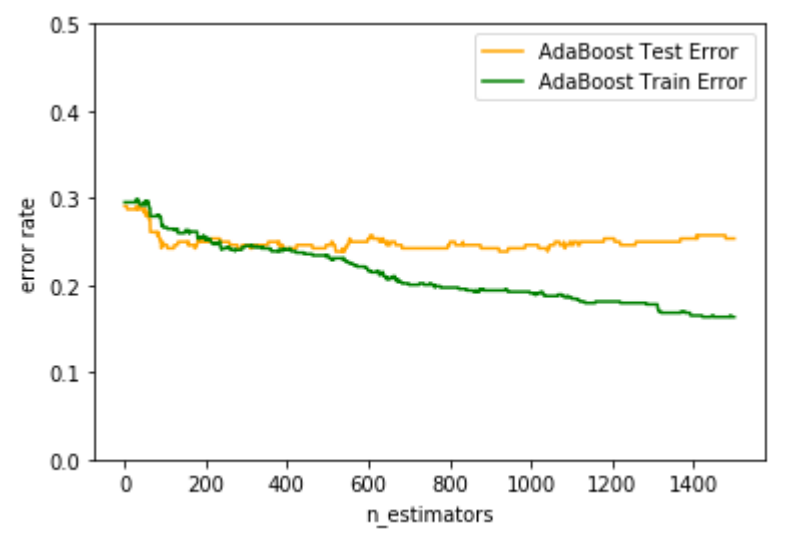
5. 의사결정 트리(Decision Tree)와 에이다부스트(AdaBoost)

- 에이다부스트(Adaboost)란 ?

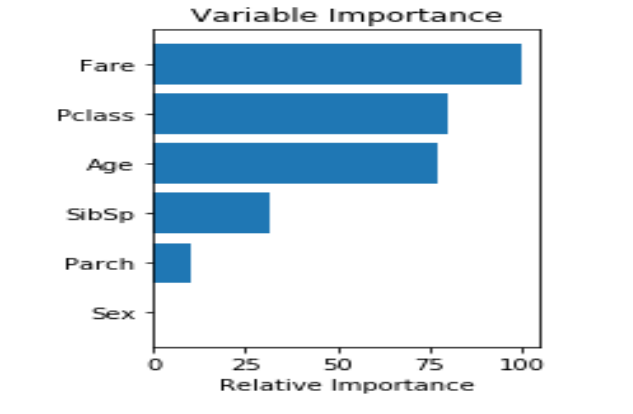
간단한 약분류기들을 상호보완 하도록 단계적으로 학습시킨 후, 이들을 조합하여 최종 강분류기의 성능을 증폭

```
DTree_train_score 0.7046548956661316
DTree_test_score 0.7089552238805971

Ada_train_score 0.8362760834670947
Ada_test_score 0.746268656716418
```



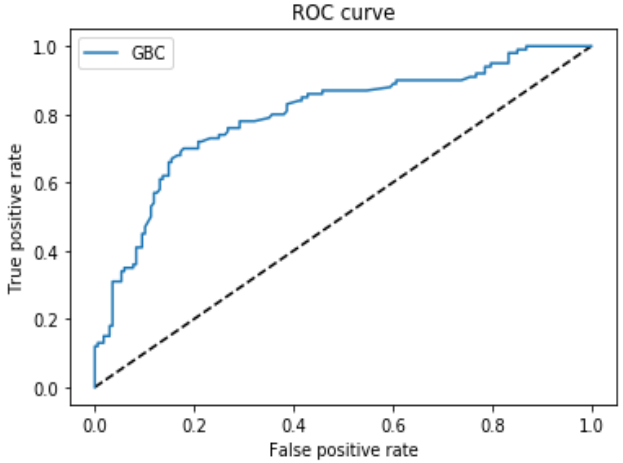
5. 그래디언트 부스팅 분류기 (GradientBoostingClassifier)



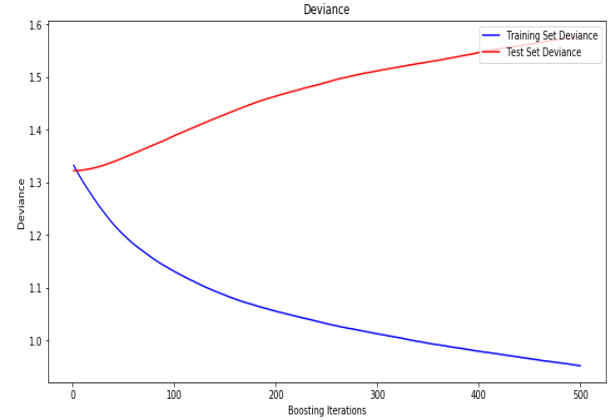
Confusion Matrix:
[[148 20]
[43 57]]
Classification Report

	precision	recall	f1-score	support
0	0.77	0.88	0.82	168
1	0.74	0.57	0.64	100

train_score: 0.7736757624398074
test_score: 0.7649253731343284



AUC: 0.7961904761904762



5. 랜덤 포레스트

3. Create model instance variable (동시에 여러 모델을 다른 이름으로 만들 수 있습니다.)

```
feature = ['Pclass', 'SibSp', 'Parch', 'Sex_clean', 'Family', 'Solo', 'Age_clean']
label = ['Survived']
```

```
from sklearn.model_selection import KFold, cross_val_score
from sklearn.ensemble import RandomForestClassifier

x = x_data[feature]
y = y_data[label]

x_train, x_test, y_train, y_test = model_selection.train_test_split(x, y, test_size=0.3, random_state=0)

k_fold = KFold(n_splits=10, shuffle=True, random_state=0)

clf = RandomForestClassifier(n_estimators=50, max_depth=4,power random_state=0)
# n_estimators == 의사결정트리의 개수
cross_val_score(clf, x_train, y_train, cv=k_fold, scoring='accuracy', ).mean()
# Accuracy
```

```
0.8282642089093702
```

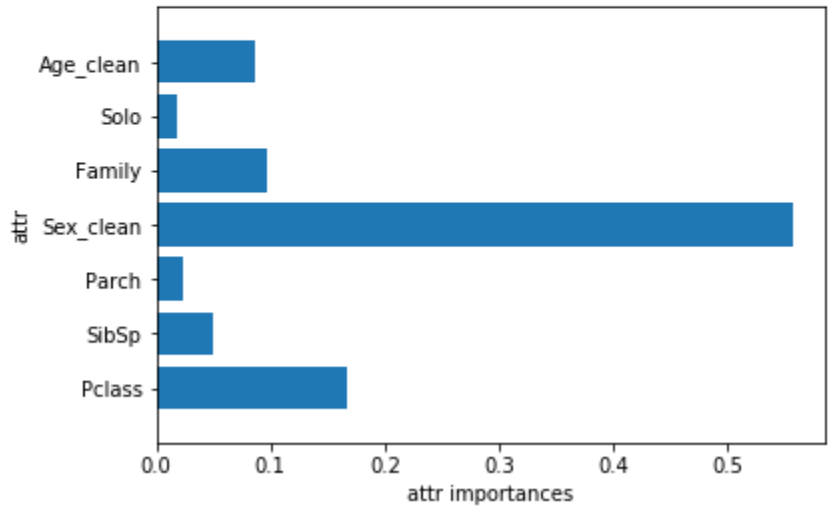
4. Train the model

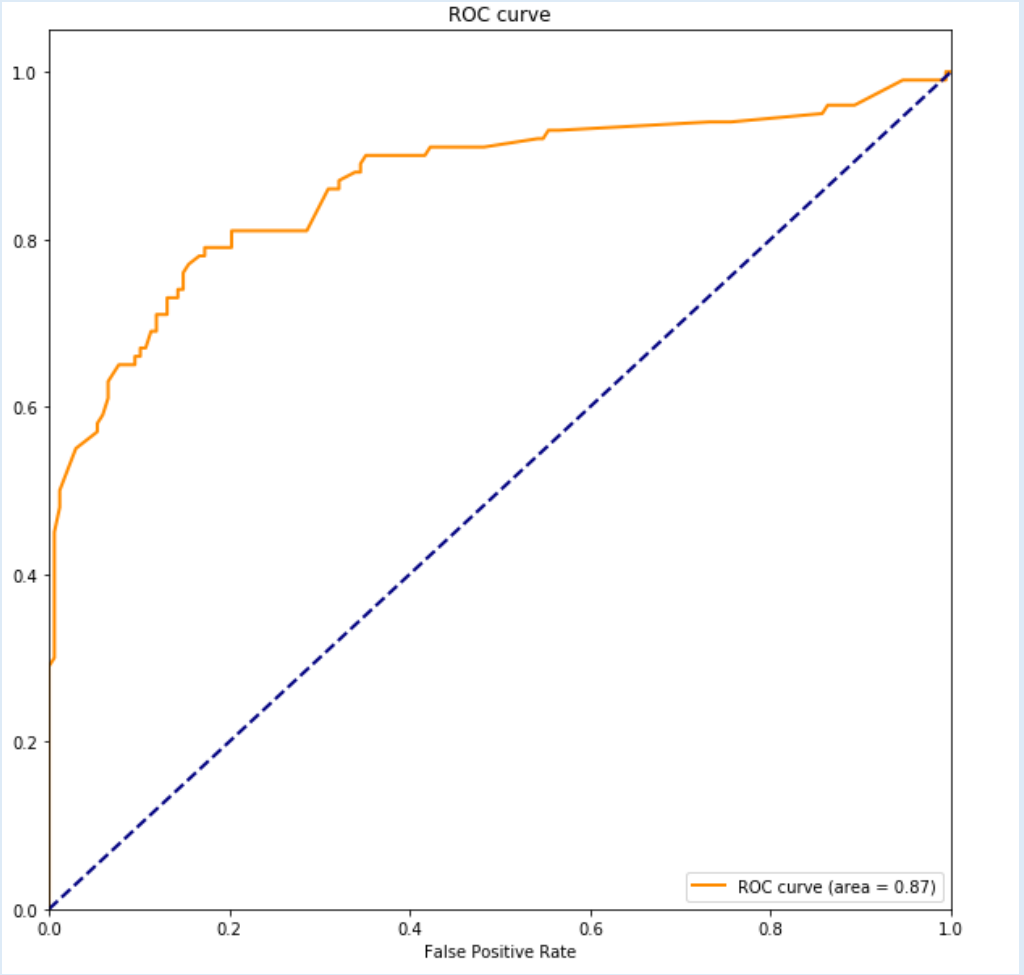
```
clf.fit(x_train,y_train)
print(clf.score(x_train,y_train))
print(clf.score(x_test,y_test))
```

```
0.8362760834670947
0.8134328358208955
```

5. 랜덤 포레스트

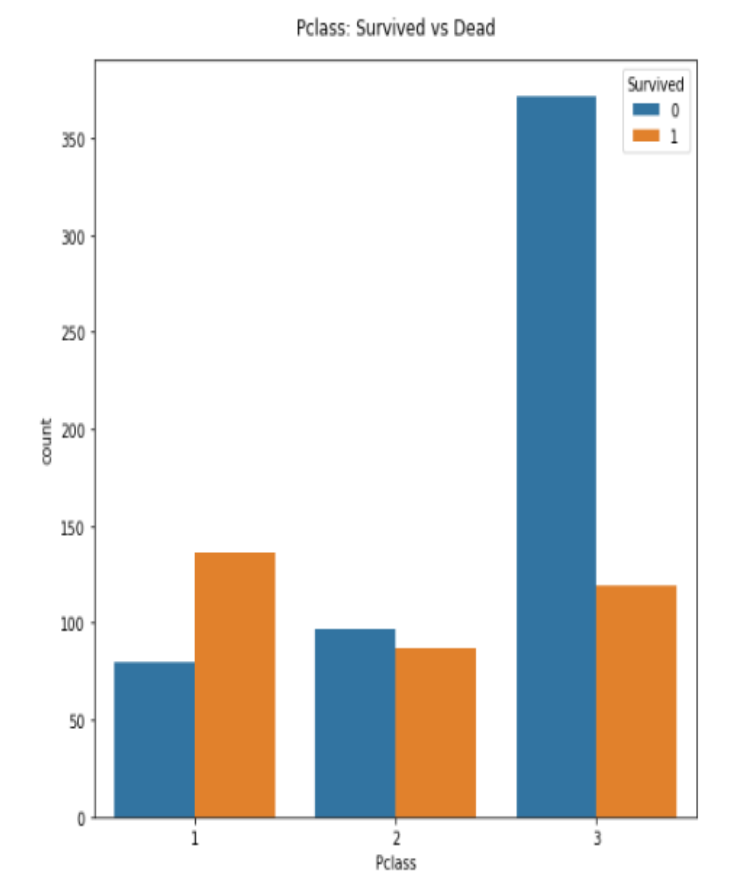
훈련 세트 정확도 : 0.836
테스트 세트 정확도 : 0.813
특성 중요도 :
[0.16710833 0.05051785 0.02346788 0.55763521 0.09779217 0.01759192
 0.08588663]

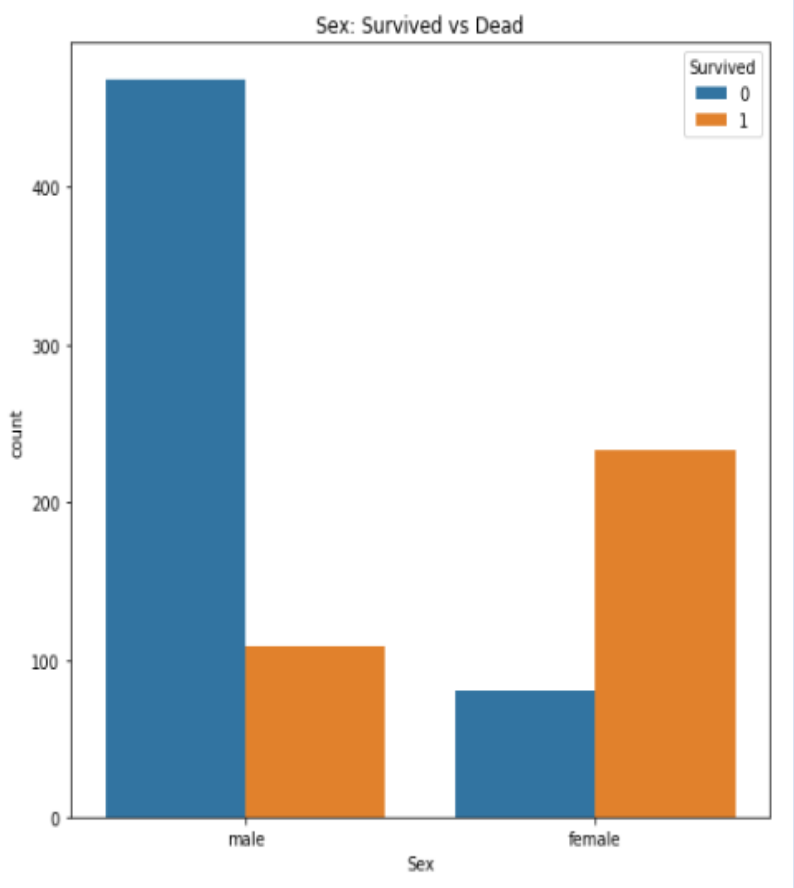


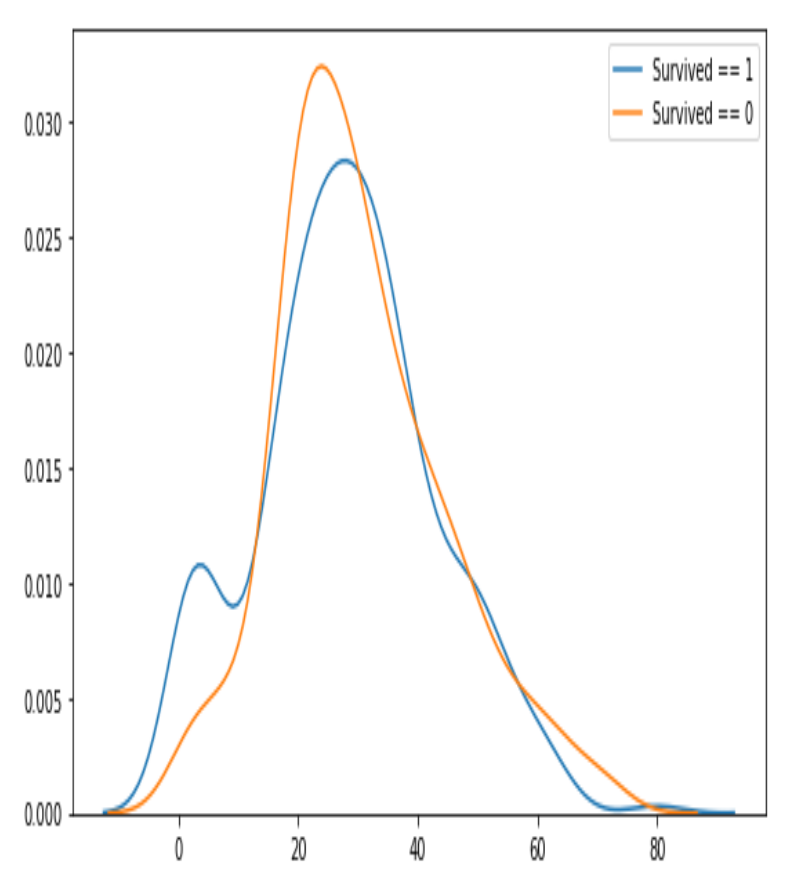


AUC 0.87

6. 데이터 분석


Pclass: Survived vs Dead
Survived
0
1
count
Pclass
1
2
3
0
50
100
150
200
250
300
350


Sex: Survived vs Dead
Survived
0
1
count
Sex
male
female
0
100
200
300
400


Survived == 1
Survived == 0
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0
20
40
60
80

7. 결론

- 1. 머신 러닝 모델 중에서 다른 모델보다 랜덤 포레스트와 서포트 벡터 머신이 점수가 높았고 두 모델 점수는 약 0.81이다.

- 2. 시작할 때 중요하다고 생각했던 열은 성별, 나이, 좌석등급이었고 모델에 따라서 중요한 열이 달랐다.

- 3. 머신러닝을 사용하지 않은 데이터 분석에서 성별과 좌석등급이 생존에 큰 영향을 미쳤다.