심장마비 : 사망원인 분석 및 예측

학과 : 컴퓨터공학전공
학번 : 2017108266
이름 : 안상민

목차

코드 소개

데이터 분석 및 시각화

예측

코드 소개

코드 소개

- DEATH_EVENT에는 여러 요소가 영향을 미침
- 이러한 여러 요소들과 DEATH_EVENT 사이의 연관성을 알아보고 이 요소들을 키 값으로 줬을 때 사망여부를 예측해보는 코드

- Age : 나이
- Sex : 성
- Anaemia : 빈혈
- Diabetes : 당뇨병
- High_blood_pressure : 고혈압
- Smoking : 흡연 여부
- DEATH_EVENT : 사망 여부

- Serum_sodium : 혈청 나트륨
- Time : 시간
- Creatinine_phosphokinase : 크레아티닌 포스포키나제
- Ejection_fraction : 구출분획
- Serum_creatinine : 혈청 크레아티닌
- Platelets : 혈소판

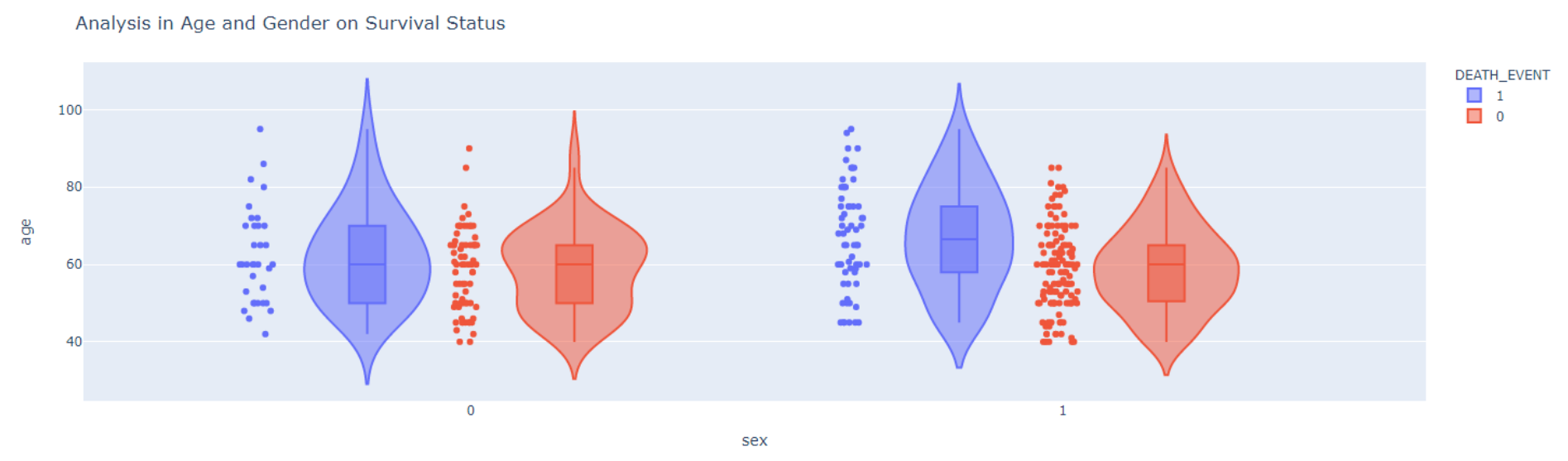
데이터 분석 및 시각화

```
heart_data = pd.read_csv('/kaggle/input/heart-failure-clinical-data/heart_failure_clinical_records_dataset.csv')
heart_data.head()
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
0	75.0	0	582	0	20	1	265000.00	1.9	130	1	0	4	1
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1	0	6	1
2	65.0	0	146	0	20	0	162000.00	1.3	129	1	1	7	1
3	50.0	1	111	0	20	0	210000.00	1.9	137	1	0	7	1
4	65.0	1	160	1	20	0	327000.00	2.7	116	0	0	8	1

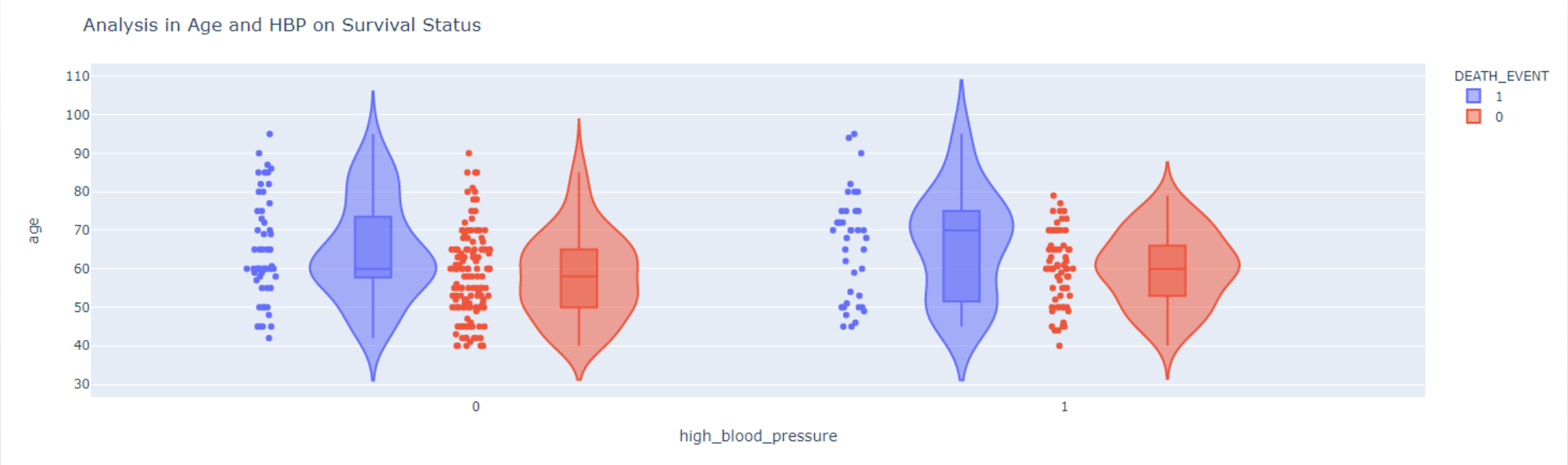
	0	1
성	여성	남성
빈혈	X	O
당뇨병	X	O
고혈압	X	O
흡연	X	O
DEATH_EVENT	생존	사망

```
fig = px.violin(heart_data, y="age", x="sex", color="DEATH_EVENT", box=True, points="all", hover_data=heart_data.columns)
fig.update_layout(title_text="Analysis in Age and Gender on Survival Status")
fig.show()
```



- 여성의 경우 50세에서 70세 사이 사람들이, 남성의 경우 60세에서 75세 사이 사람들이 대부분 사망

```
fig = px.violin(heart_data, y="age", x="high_blood_pressure", color="DEATH_EVENT", box=True, points="all", hover_data=heart_data.columns)
fig.update_layout(title_text="Analysis in Age and HBP on Survival Status")
fig.show()
```

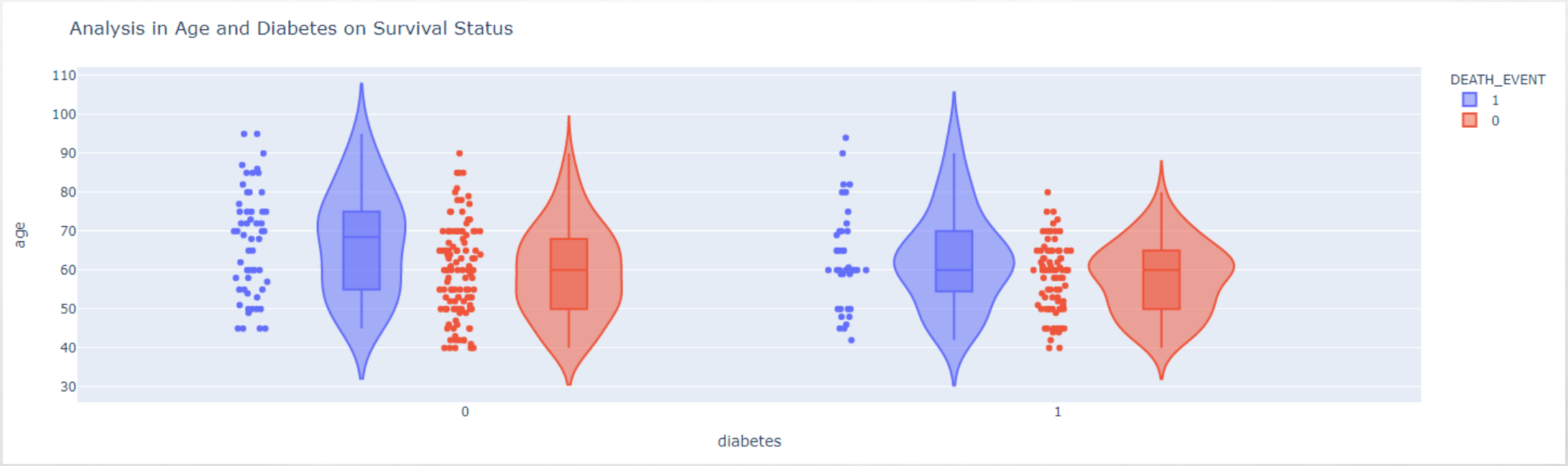


- 고혈압인 사람의 경우 50세에서 75세 사이 사람들이, 고혈압이 아닌 사람의 경우 60세에서 75세 사이 사람들이 대부분 사망

```
fig = px.violin(heart_data, y="age", x="smoking", color="DEATH_EVENT", box=True, points="all", hover_data=heart_data.columns)
fig.update_layout(title_text="Analysis in Age and Smoking on Survival Status")
fig.show()
```

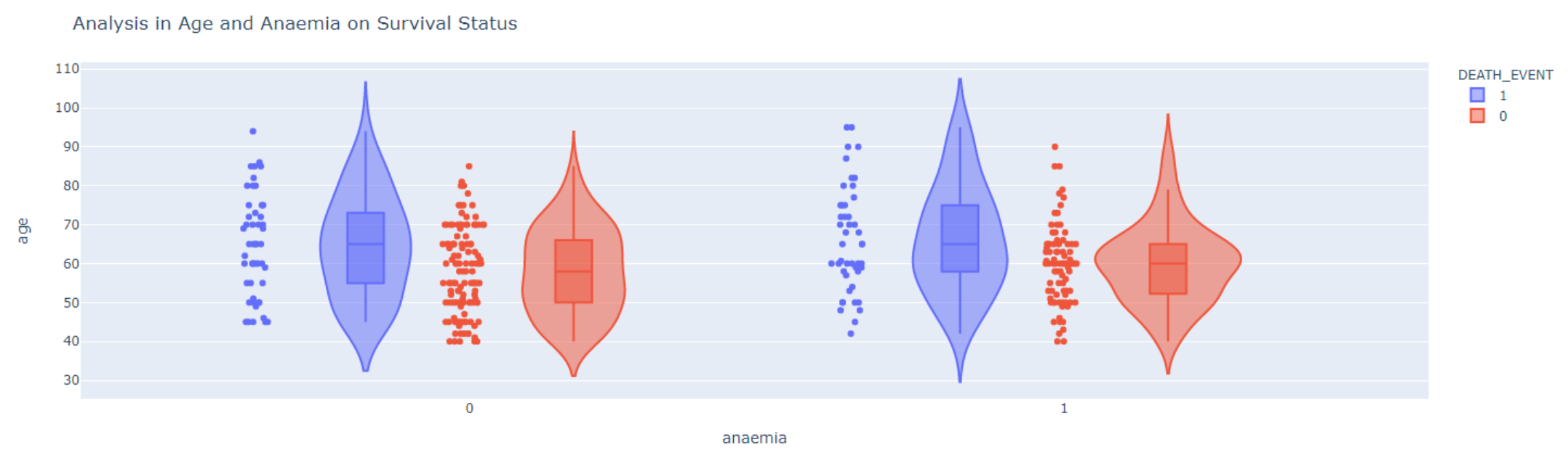
- 비흡연자의 경우 50세에서 75세 사이 사람들이, 흡연자의 경우 60세에서 70세 사이 사람들이 대부분 사망

```
fig = px.violin(heart_data, y="age", x="diabetes", color="DEATH_EVENT", box=True, points="all", hover_data=heart_data.columns)
fig.update_layout(title_text="Analysis in Age and Diabetes on Survival Status")
fig.show()
```



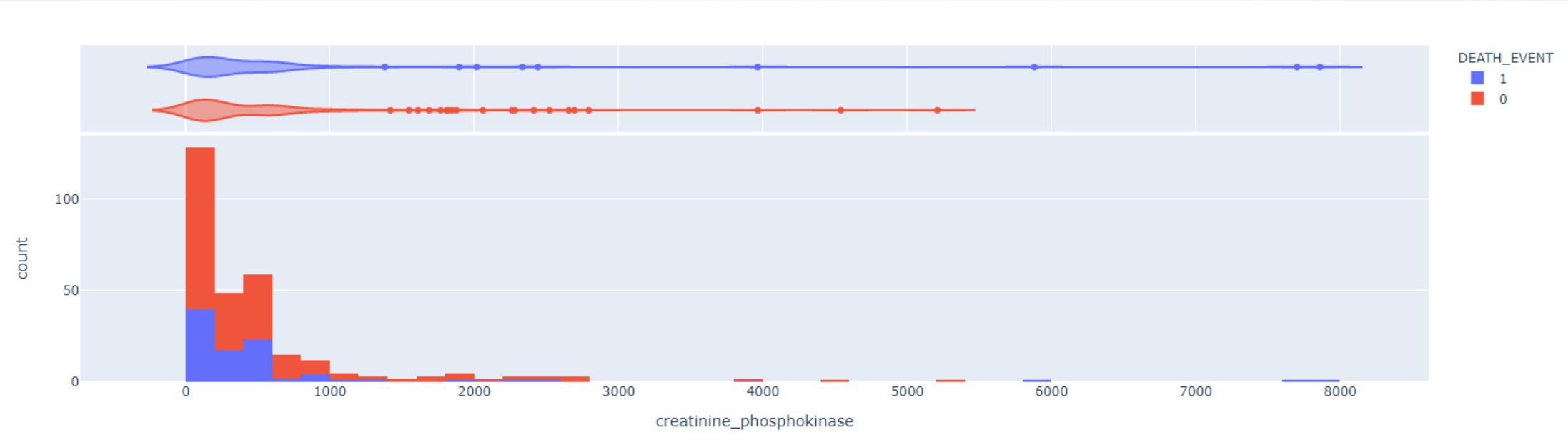
- 당뇨병에 걸리지 않은 사람들은 55세에서 75세 사이 사람들이,
  당뇨병에 걸린 사람들은 55세에서 70세 사이 사람들이 대부분 사망

```
fig = px.violin(heart_data, y="age", x="anaemia", color="DEATH_EVENT", box=True, points="all", hover_data=heart_data.columns)
fig.update_layout(title_text="Analysis in Age and Anaemia on Survival Status")
fig.show()
```



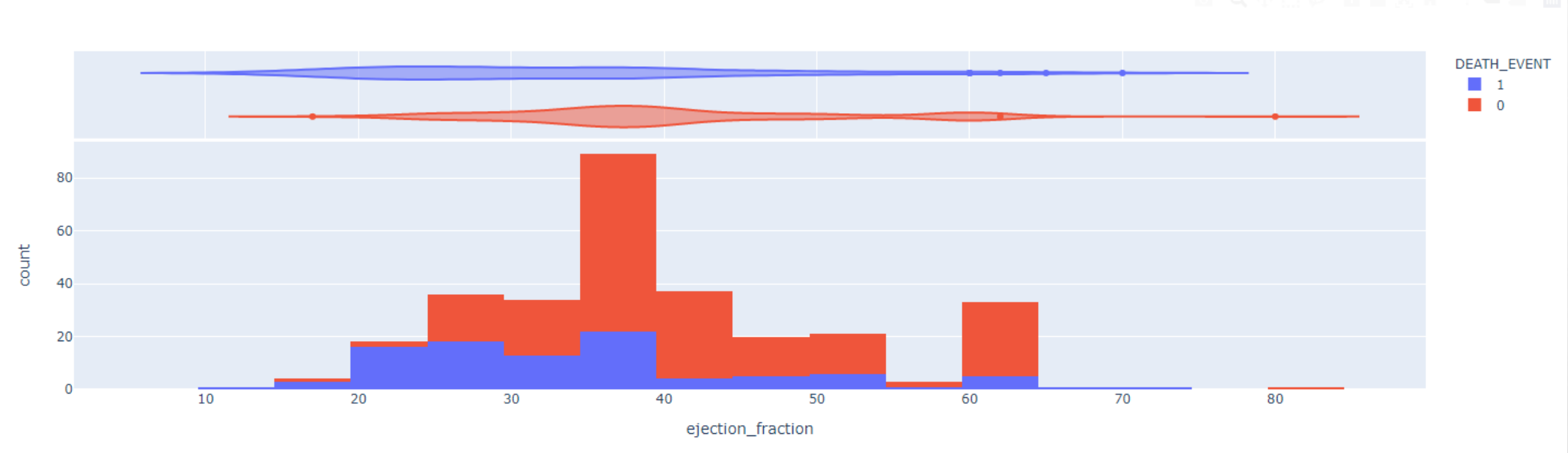
- 빈혈이 아닌 사람들은 55세에서 75세 사이 사람들이, 빈혈인 사람들은 55세에서 70세 사이 사람들이 대부분 사망

```
fig = px.histogram(heart_data, x="creatinine_phosphokinase", color="DEATH_EVENT", marginal="violin", hover_data=heart_data.columns)
fig.show()
```



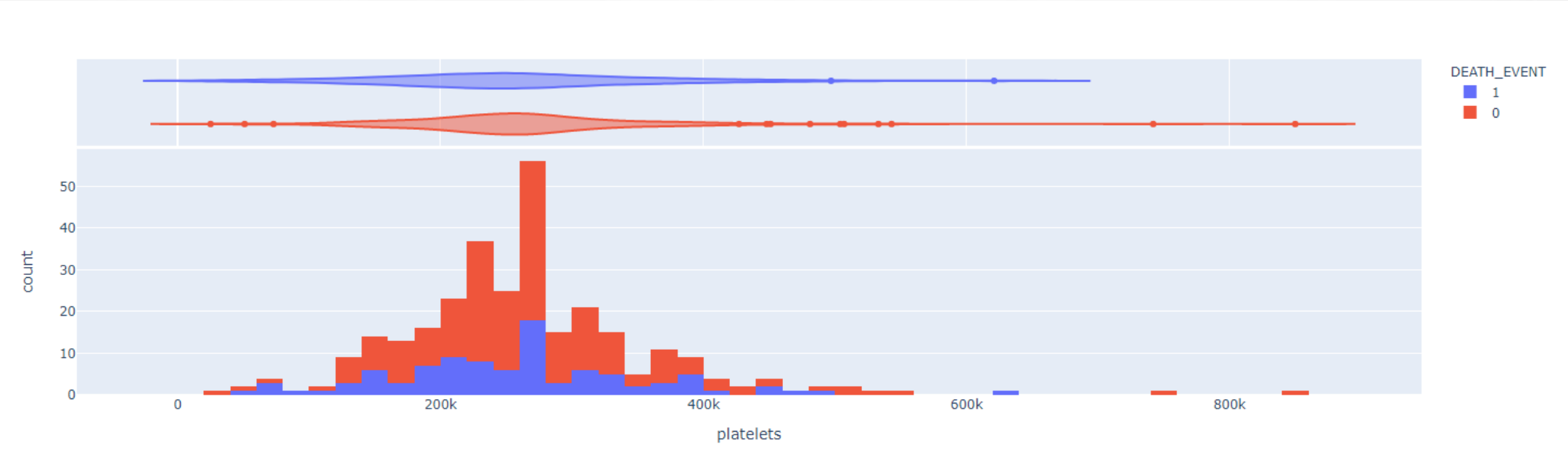
- 크레아티닌 포스포키나제의 수치가 0~100일 때 사망자와 생존자의 수치가 높게 나타남

```
fig = px.histogram(heart_data, x="ejection_fraction", color="DEATH_EVENT", marginal="violin", hover_data=heart_data.columns)
fig.show()
```



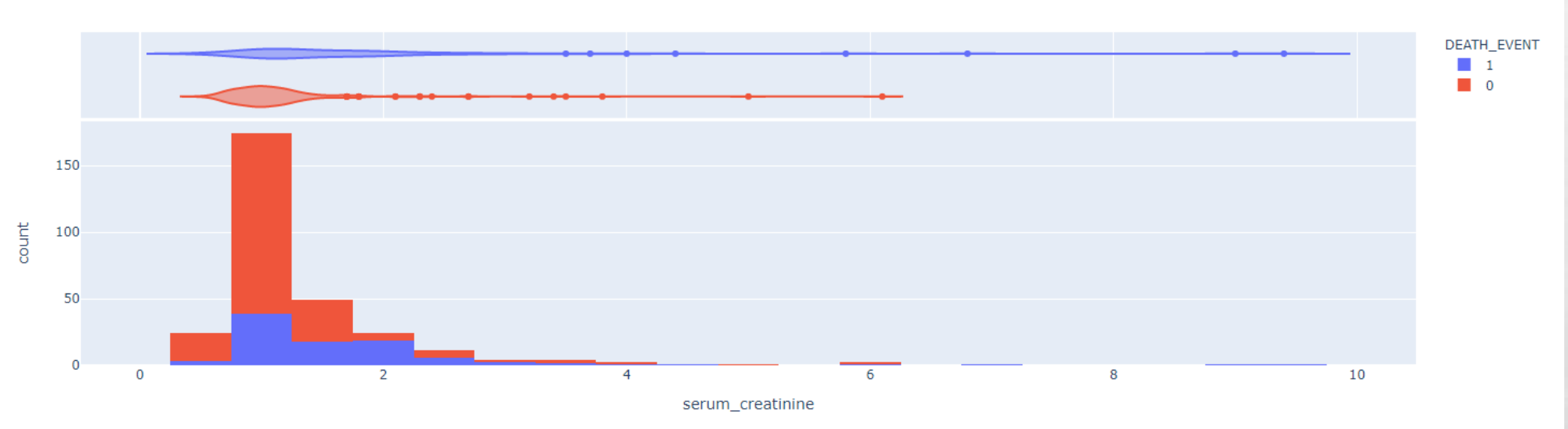
- 구출분획 수치가 35~40일 때 사망자와 생존자의 수치가 높게 나타남

```
fig = px.histogram(heart_data, x="platelets", color="DEATH_EVENT", marginal="violin", hover_data=heart_data.columns)
fig.show()
```



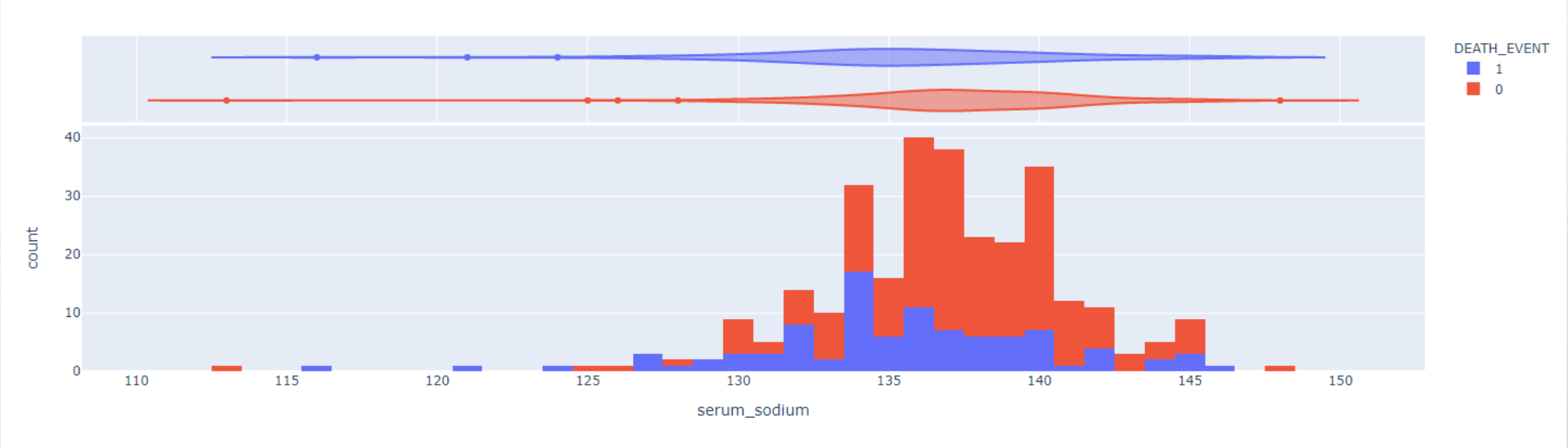
- 혈소판 수치가 260k~280k일 때 사망자와 생존자의 수치가 높게 나타남

```
fig = px.histogram(heart_data, x="serum_creatinine", color="DEATH_EVENT", marginal="violin", hover_data=heart_data.columns)
fig.show()
```



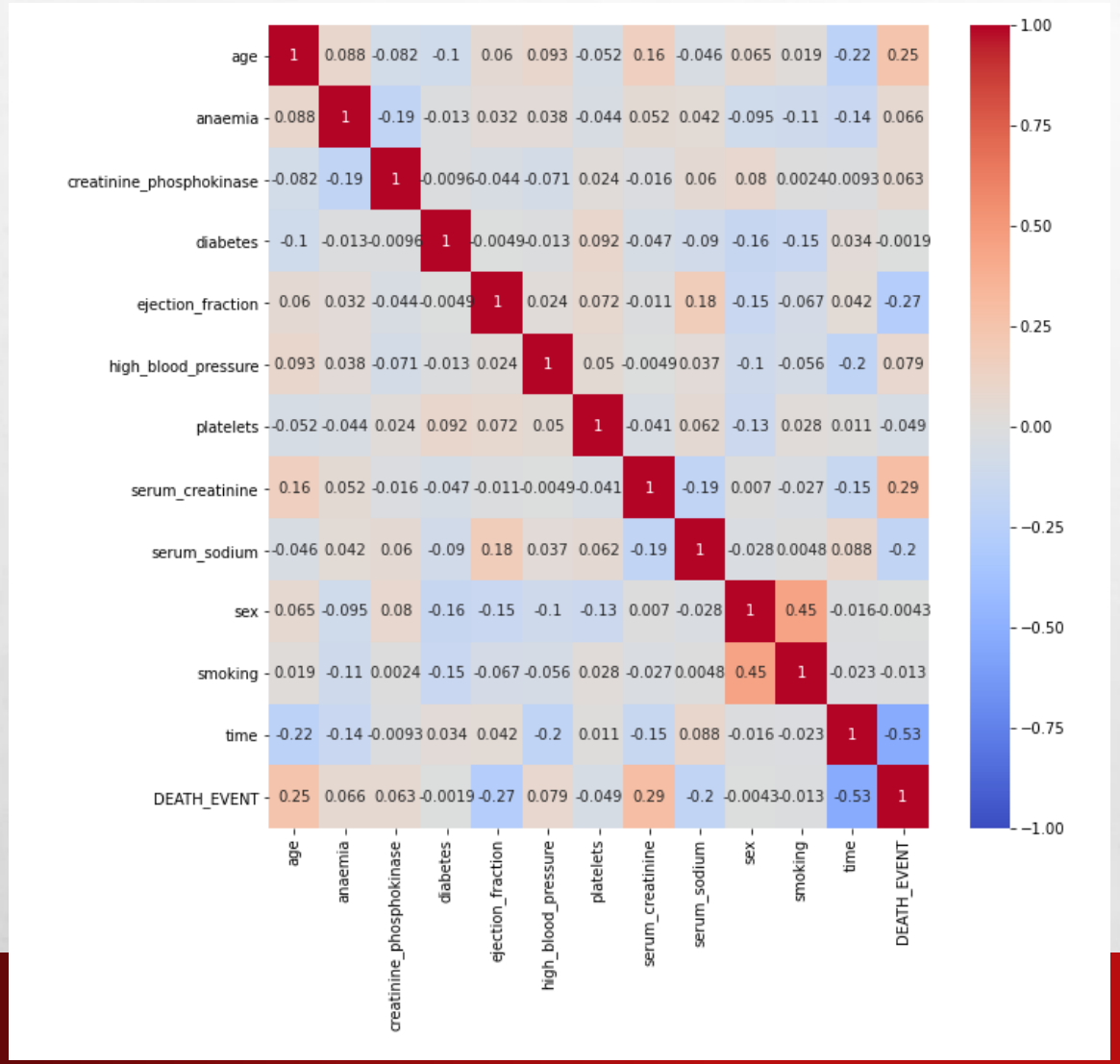
- 혈청 크레아티닌 수치가 0.75~1.25일 때 사망자와 생존자의 수치가 높게 나타남

```
fig = px.histogram(heart_data, x="serum_sodium", color="DEATH_EVENT", marginal="violin",hover_data=heart_data.columns)
fig.show()
```



- 혈청 나트륨 수치가 136일때 생존자의 수치가 제일 높고 134일때 사망자의 수치가 제일 높게 나타남

```
plt.figure(figsize=(10,10))
sns.heatmap(heart_data.corr(), vmin=-1, cmap='coolwarm', annot=True);
```



- X축의 요소와 Y축의 요소의 상관관계를 보여줌.
- DEATH_EVENT랑 상관관계가 가장 큰 요소는 혈청 크레아티닌
- 상관관계가 가장 작은 요소는 시간

예측

```
Features = ['time','ejection_fraction','serum_creatinine']
x = heart_data[Features]
y = heart_data["DEATH_EVENT"]
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size=0.2, random_state=2)
```

```
accuracy_list = []
```

- Time, ejection_fraction, serum_creatinine 이 3가지 키 값에 따른 DEATH_EVENT를 예측
- 전체 데이터에서 80%를 학습용, 20%를 테스트용으로 나눔
- 여러 예측 알고리즘을 통해 나온 정확도를 accuracy_list에 저장

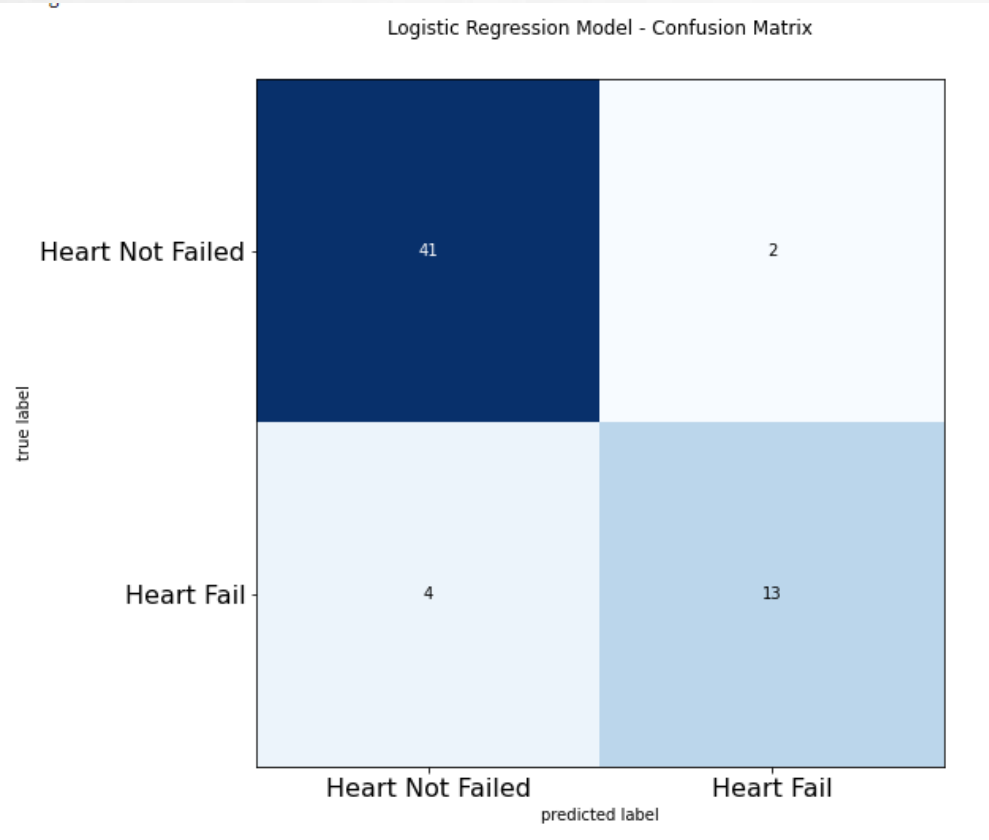
```
# logistic regression

log_reg = LogisticRegression()
log_reg.fit(x_train, y_train)
log_reg_pred = log_reg.predict(x_test)
log_reg_acc = accuracy_score(y_test, log_reg_pred)
accuracy_list.append(100*log_reg_acc)
```

- LogisticRegression을 사용해 데이터 학습
- 테스트 문제를 줘서 예측
- 예측을 해서 나온 정확성은 accuracy_list에 저장

```
print(Fore.GREEN + "Accuracy of Logistic Regression is : ", "{:.2f}%".format(100* log_reg_acc))
```

```
Accuracy of Logistic Regression is :  90.00%
```

```
cm = confusion_matrix(y_test, log_reg_pred)
plt.figure()
plot_confusion_matrix(cm, figsize=(12,8), hide_ticks=True, cmap=plt.cm.Blues)
plt.title("Logistic Regression Model - Confusion Matrix")
plt.xticks(range(2), ["Heart Not Failed","Heart Fail"], fontsize=16)
plt.yticks(range(2), ["Heart Not Failed","Heart Fail"], fontsize=16)
plt.show()
```

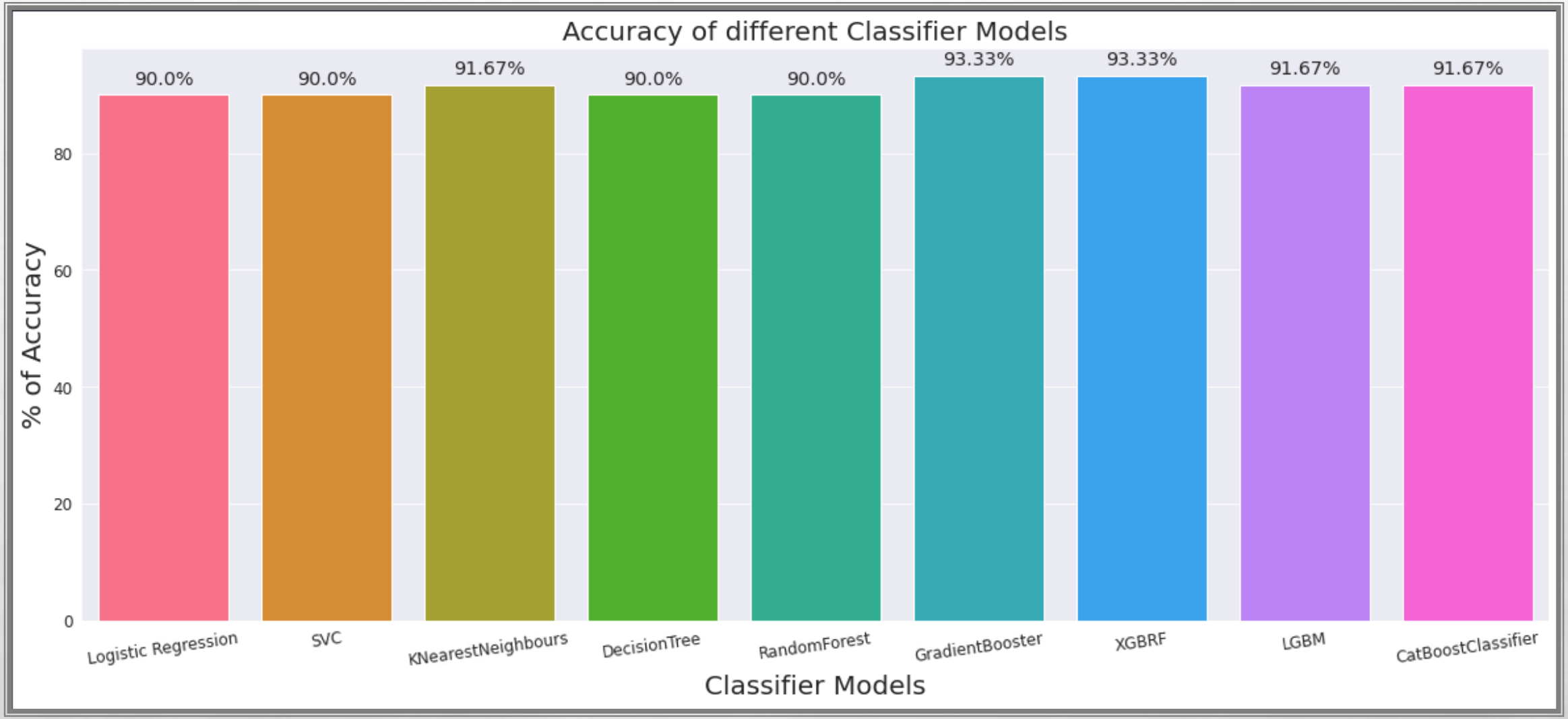


- PLOT_CONFUSION_MATRIX 사용 -> 좌상단에서 우하단으로 가는 대각선의 값이 높을수록 예측이 잘 되는 것
- 예측해서 맞춘 값의 수 = 41+13 = 54
- 예측해서 맞추지 못한 값의 수 = 4+2 = 6
- 맞춘 값의 수/전체 데이터 수 = (54/60)*100 = 90%

```
model_list = ['Logistic Regression', 'SVC','KNearestNeighbours', 'DecisionTree', 'RandomForest',
              'GradientBooster', 'XGBRF','LGBM', 'CatBoostClassifier']
```

```
plt.rcParams['figure.figsize']=20,8
sns.set_style('darkgrid')
ax = sns.barplot(x=model_list, y=accuracy_list, palette = "husl", saturation =2.0)
plt.xlabel('Classifier Models', fontsize = 20 )
plt.ylabel('% of Accuracy', fontsize = 20)
plt.title('Accuracy of different Classifier Models', fontsize = 20)
plt.xticks(fontsize = 12, horizontalalignment = 'center', rotation = 8)
plt.yticks(fontsize = 12)
for i in ax.patches:
    width, height = i.get_width(), i.get_height()
    x, y = i.get_xy()
    ax.annotate(f'{round(height,2)}%', (x + width/2, y + height*1.02), ha='center', fontsize = 'x-large')
plt.show()
```

- 모델별 정확성을 비교,시각화


Accuracy of different Classifier Models
% of Accuracy
0
20
40
60
80
90.0%
90.0%
91.67%
90.0%
90.0%
93.33%
93.33%
91.67%
91.67%
Logistic Regression
SVC
KNearestNeighbours
DecisionTree
RandomForest
GradientBooster
XGBRF
LGBM
CatBoostClassifier
Classifier Models

Kaggle 링크 : https://www.kaggle.com/nayansakhiya/heart-fail-analysis-and-quick-prediction