2023-2 KUBIG

Drug Classification 프로젝트

ML 분반 2팀
이준복 신인수 진서연

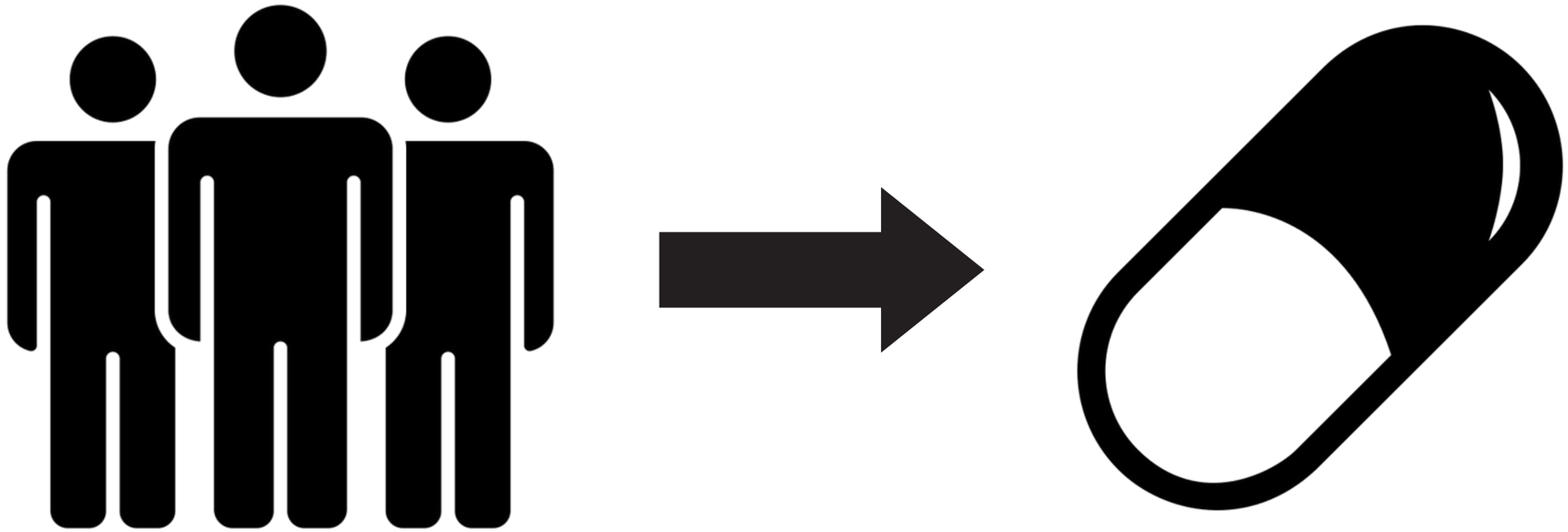
목차

1

프로젝트 목표

Part 1

프로젝트 목표



사람의 건강상태 정보가 주어진 상황에서 사용한 Drug를 예측하는 문제
Supervised learning, Classification problem

2

EDA

2.1 변수 설명

수치형 (Numerical)	범주형 (Categorical)
• Age (Age of person) : integer • Na_to_K (Ratio of sodium and potassium) : float	• BP (Blood pressure level) : High, Normal, Low • Cholesterol (Cholesterol level) : High, Normal • Sex (Gender of person) : M, F • Drug: (Type of drug used) A, B, C, X, Y → **Target**

2.1 변수 설명

info, describe 함수를 통해 6개 변수의 데이터 정보 확인

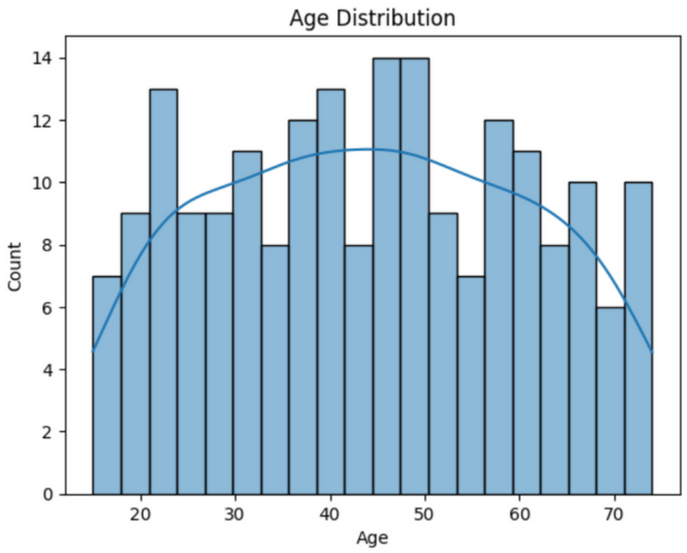
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Age          200 non-null    int64  
 1   Sex          200 non-null    object 
 2   BP           200 non-null    object 
 3   Cholesterol  200 non-null    object 
 4   Na_to_K      200 non-null    float64
 5   Drug         200 non-null    object 
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
count	200.000000	200	200	200	200.000000	200
unique	NaN	2	3	2	NaN	5
top	NaN	M	HIGH	HIGH	NaN	DrugY
freq	NaN	104	77	103	NaN	91
mean	44.315000	NaN	NaN	NaN	16.084485	NaN
std	16.544315	NaN	NaN	NaN	7.223956	NaN
min	15.000000	NaN	NaN	NaN	6.269000	NaN
25%	31.000000	NaN	NaN	NaN	10.445500	NaN
50%	45.000000	NaN	NaN	NaN	13.936500	NaN
75%	58.000000	NaN	NaN	NaN	19.380000	NaN
max	74.000000	NaN	NaN	NaN	38.247000	NaN

데이터 수 : 200개
수치형 변수 : 2개
범주형 변수 : 3개

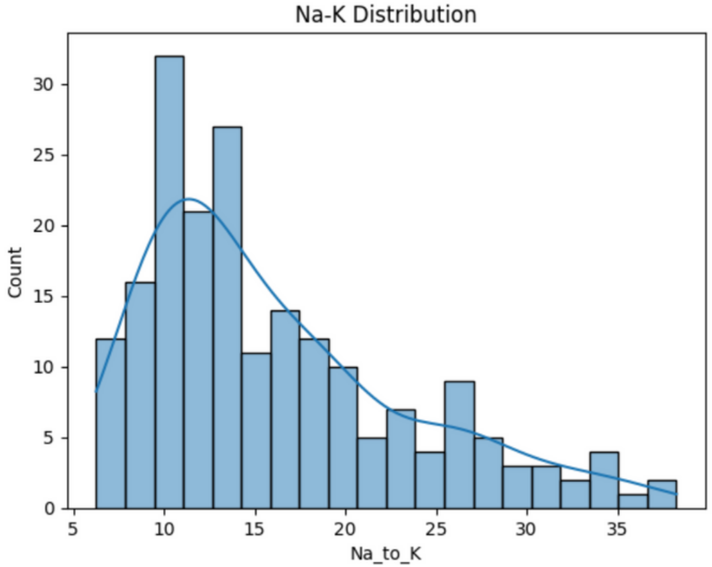
2.2 수치형 변수 EDA

Age (연령)



연령대는 포물선 형태의 데이터 분포를 띠고 있음

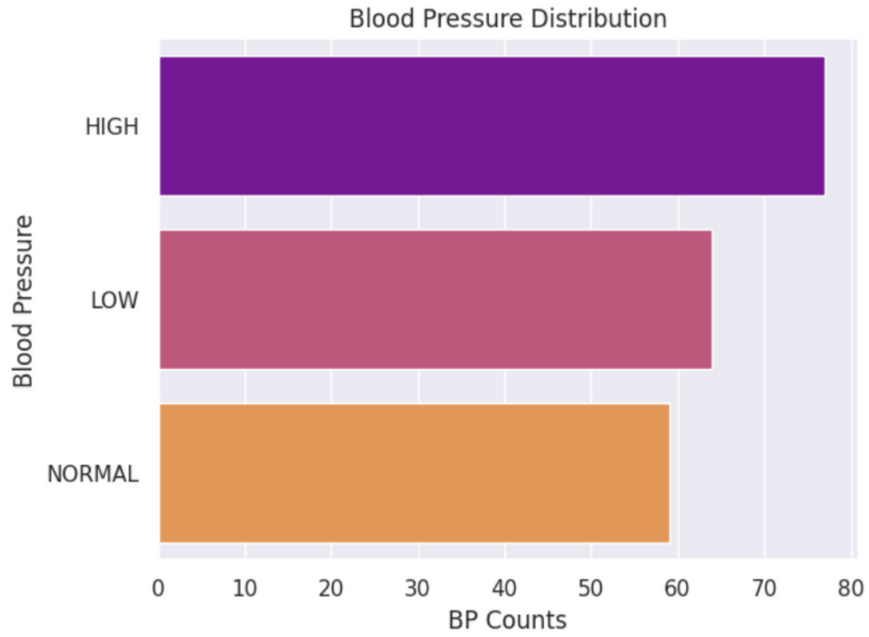
Na_to_K (나트륨과 칼륨의 비율)



Na_to_K는 비교적 낮은 값의 데이터가 많은 분포를 지님

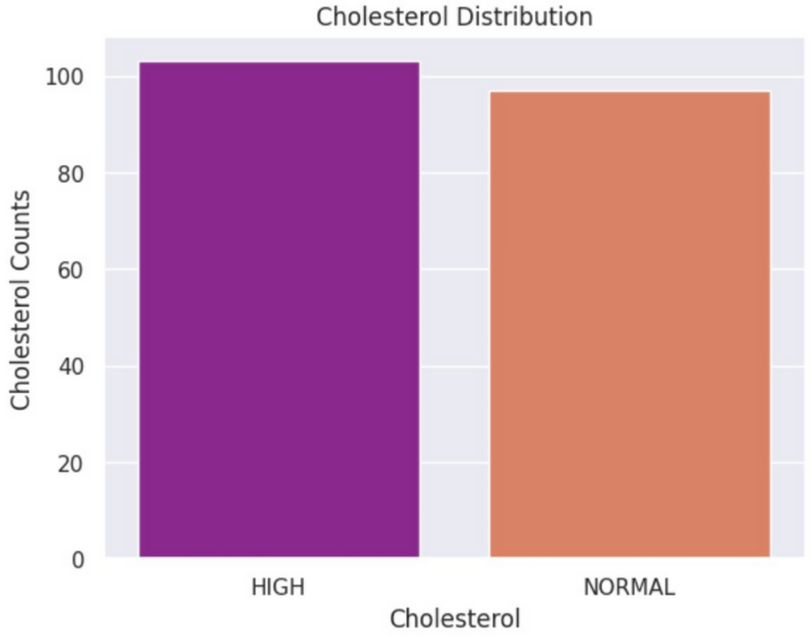
2.3 범주형 변수 EDA

BP
(혈압 수치)



혈압 수치는 HIGH > LOW > NORMAL 순서로 데이터가 많음

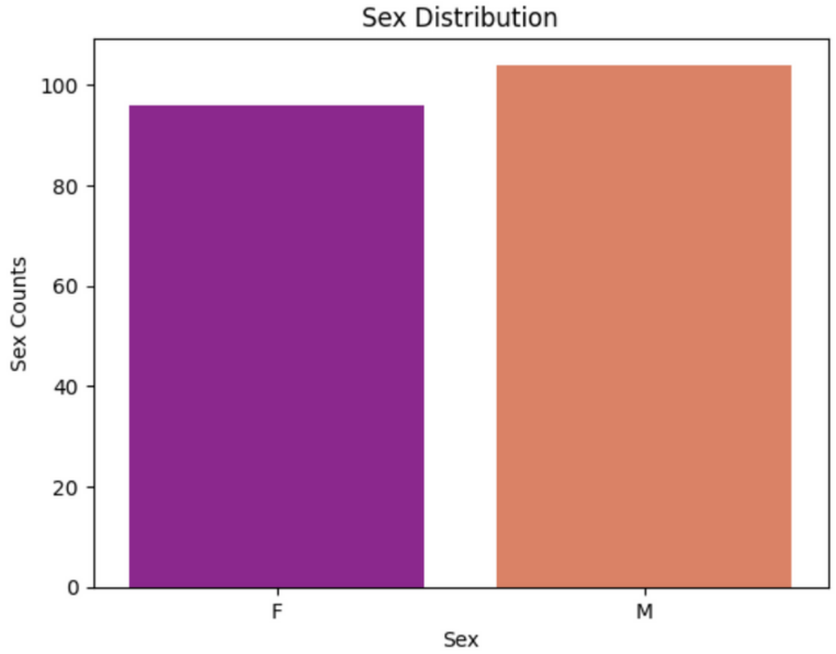
Cholesterol
(콜레스테롤)



콜레스테롤 수치는 HIGH와 NORMAL이
비슷한 수치만큼 존재함

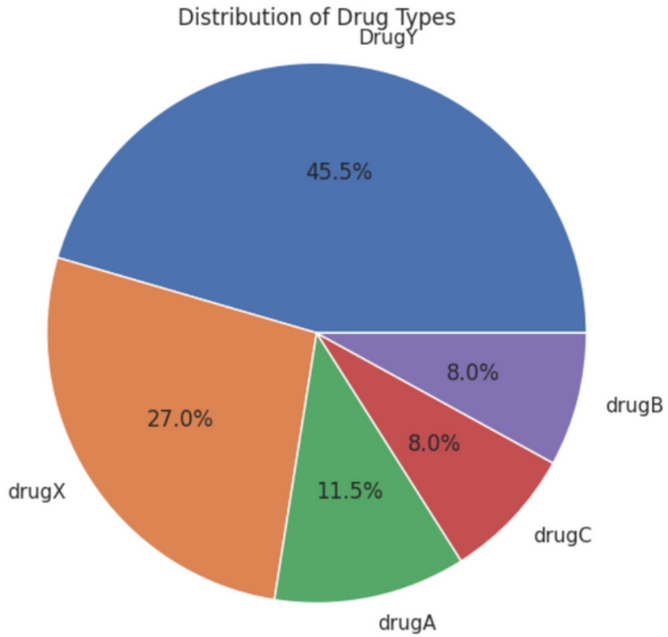
2.3 범주형 변수 EDA

Sex (성별)



성별도 남/녀 비슷한 수치를 보여주고 있음

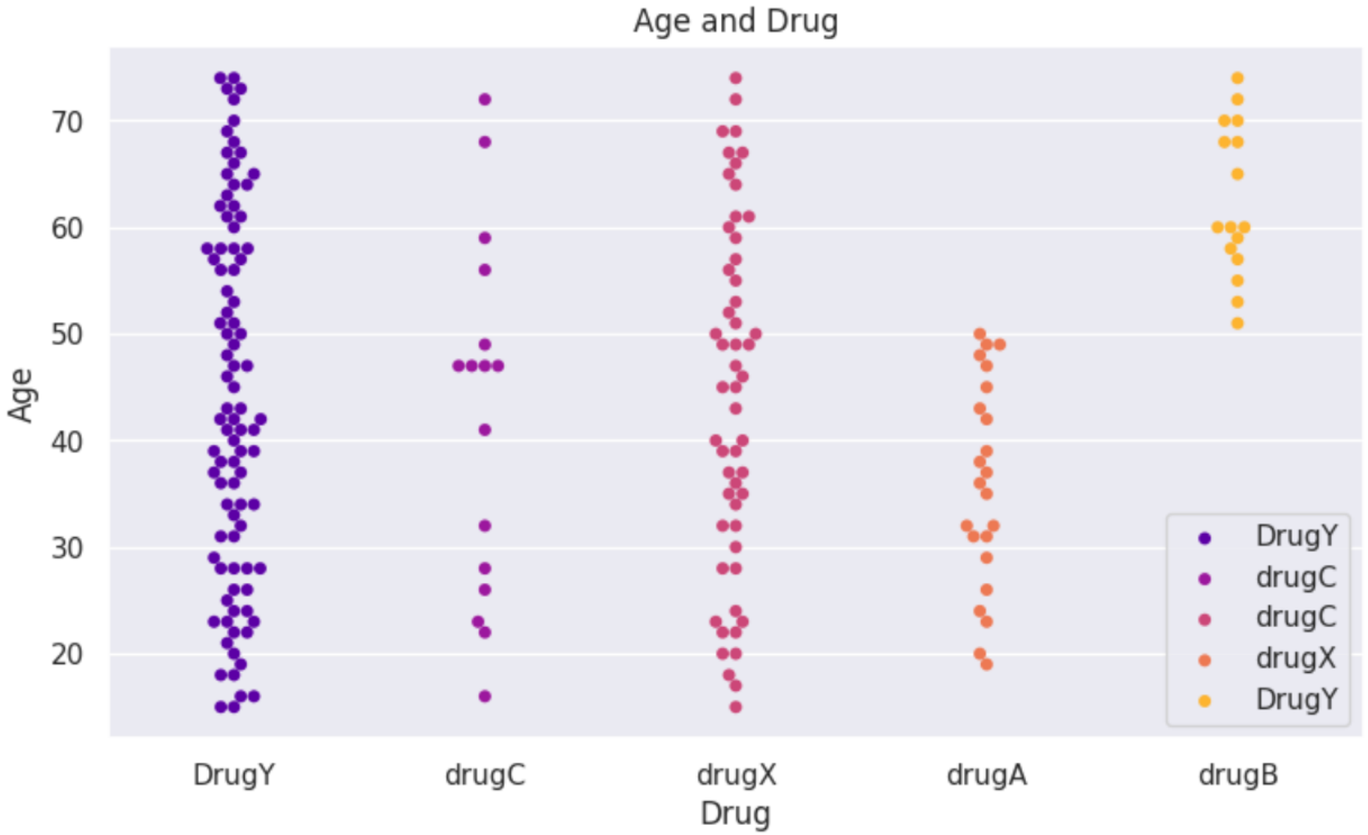
Drug (의약품)



종속변수 Drug는 Y > X > A > C > B 순서로 데이터가 많음

2.4 독립변수 <-> 종속변수 데이터 분포 확인

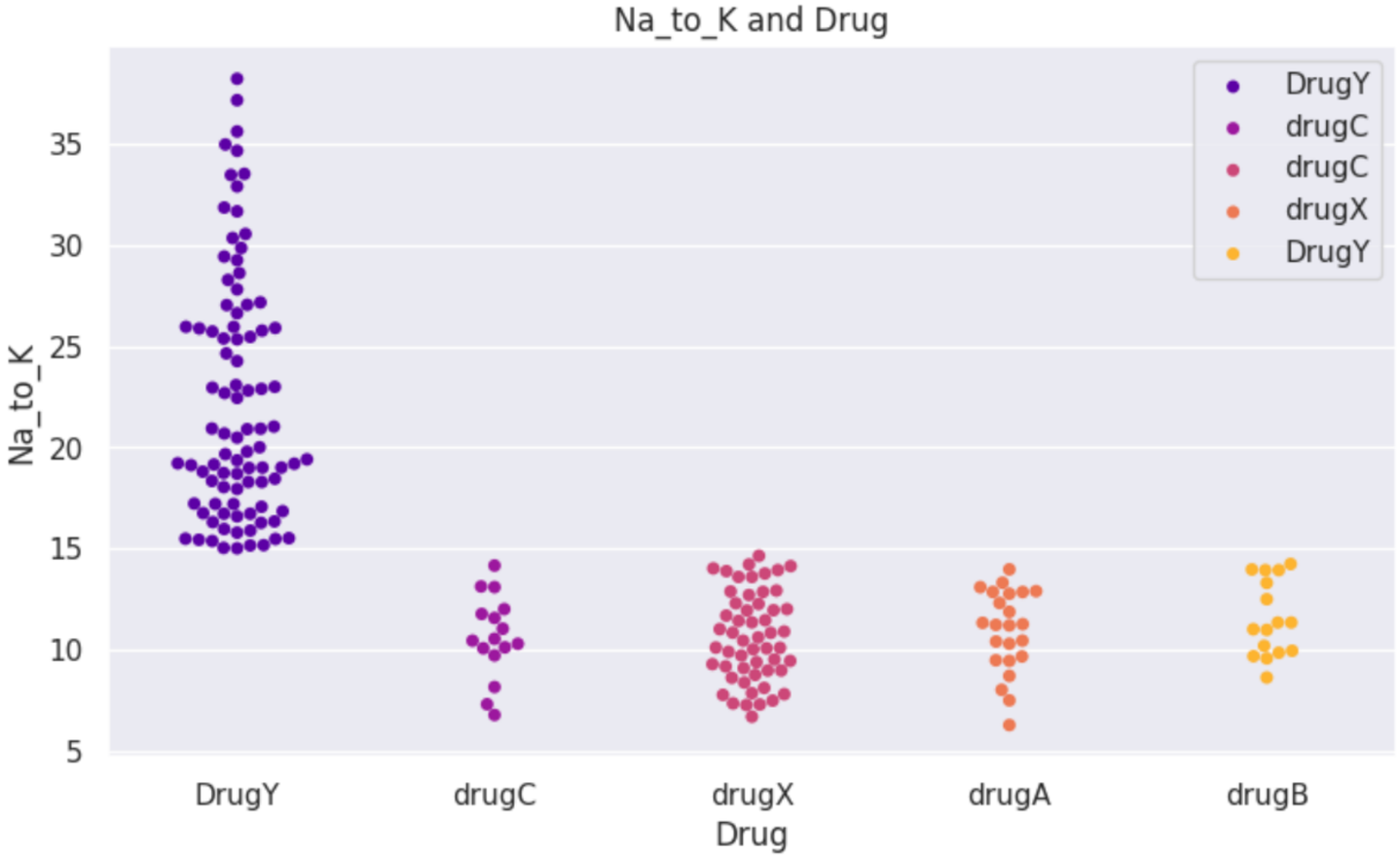
Age <-> Drug



Drug B는 나이가 50 이상인 경우, Drug A는 50 이하인 경우에만 처방. 그 외에 3개의 유형은 모든 연령대에 대한 값을 가짐
>> Age 변수는 Drug 분류에 영향을 미칠 것으로 예상

2.4 독립변수 <-> 종속변수 데이터 분포 확인

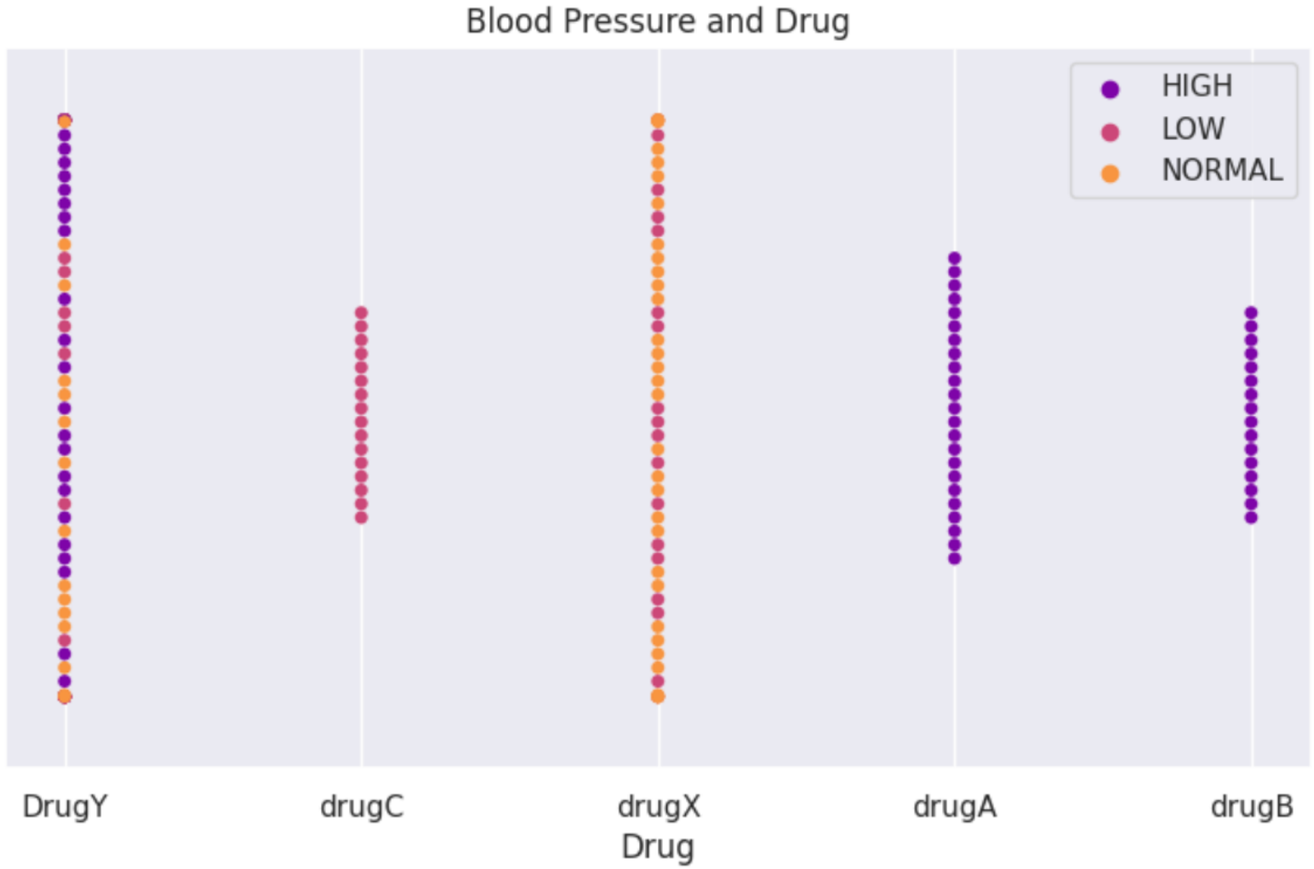
Na_to_K <-> Drug



DrugY는 Na_to_K가 15 이상인 경우, 그 이외의 약 유형은 15 이하인 경우에 처방
>> Na_to_K 변수는 Drug 분류에 영향을 미칠 것으로 예상

2.4 독립변수 <-> 종속변수 데이터 분포 확인

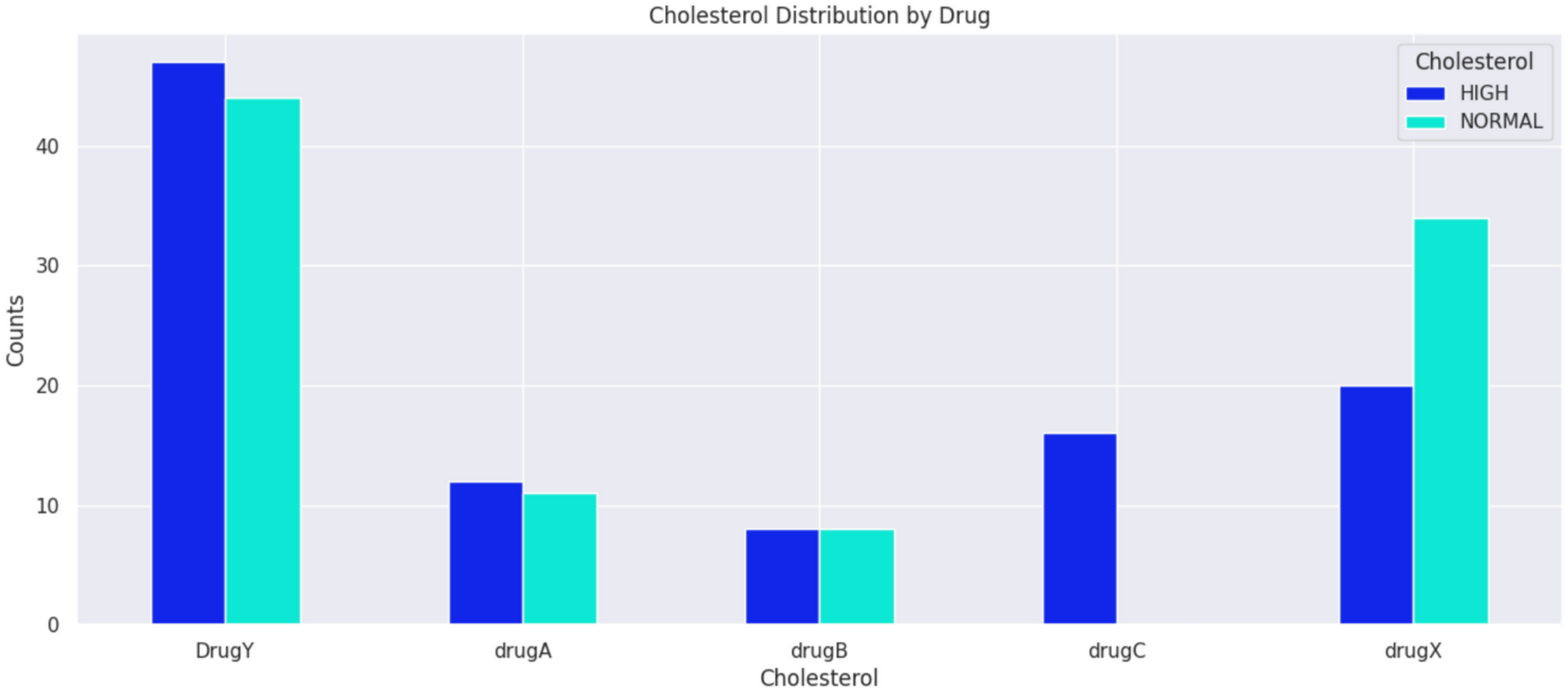
BP <-> Drug



Drug Y만 모든 혈압에 사용 가능하고, 그 외 4개의 Drug는 특정 혈압 수준에만 사용 가능함
>> BP 변수는 Drug 분류에 영향을 미칠 것으로 예상

2.4 독립변수 <-> 종속변수 데이터 분포 확인

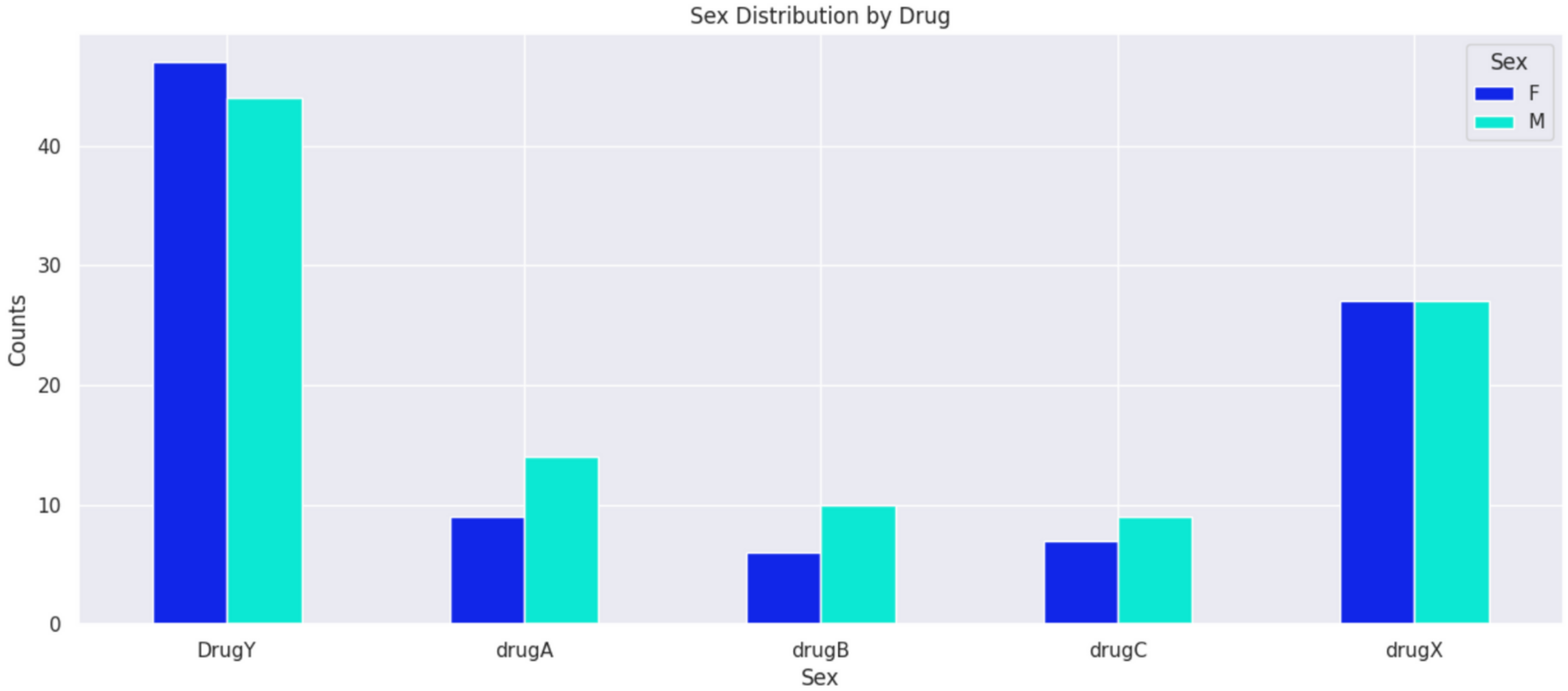
Cholesterol <-> Drug



Drug C는 높은 콜레스테롤 수치에만 처방 가능
>>Cholesterol 변수는 Drug 분류에 영향을 미칠 것으로 예상

2.4 독립변수 <-> 종속변수 데이터 분포 확인

Sex <-> Drug



성별에 따라 약의 유형이 달라지진 않음
>> 다른 4개의 변수와 달리, 성별 변수는 Drug 분류에 직접적인 영향을 미치지 않을 것
>> 최종 결과와 비교 예정

2.5 수치형 변수 - 정규성 검정

Shapiro Wilk Test

H_0 : Distribution follows normal

H_1 : Distribution does not follows normal

```
# Shapiro wilk test Age
x = drug["Age"].to_numpy()
stats.shapiro(x.reshape(1, -1))
```

```
ShapiroResult(statistic=0.9639396071434021, pvalue=5.4086522141005844e-05)
```

```
# Shapiro wilk test Na_to_K
x = drug["Na_to_K"].to_numpy()
stats.shapiro(x.reshape(1, -1))
```

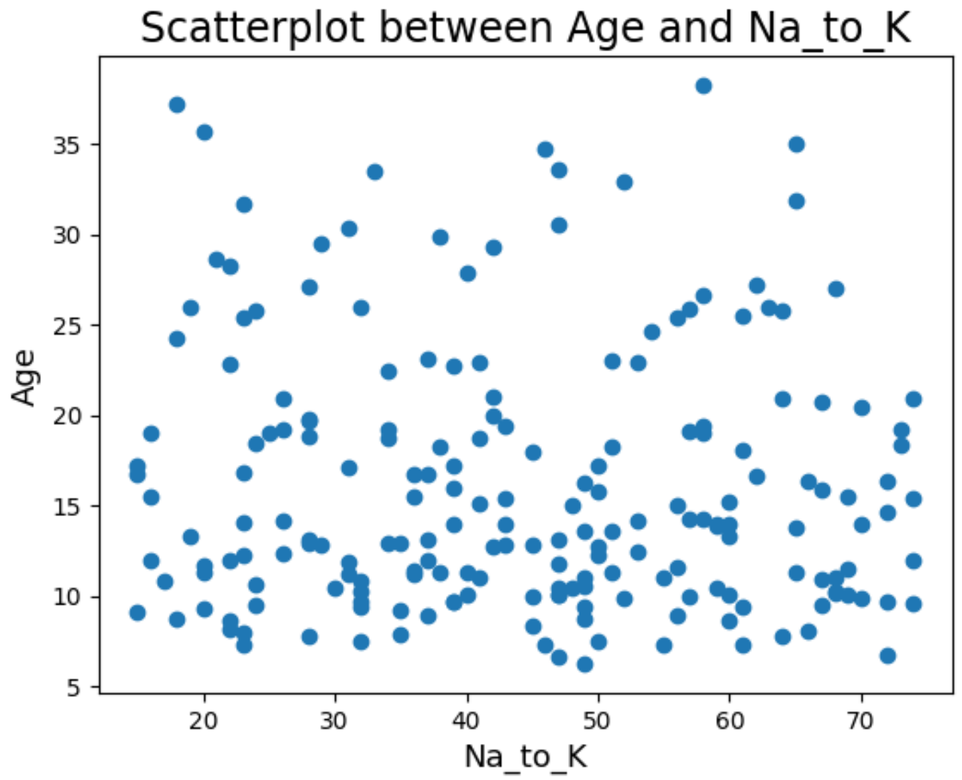
```
ShapiroResult(statistic=0.901857852935791, pvalue=3.305025975119946e-10)
```

Rejects null hypothesis at α = 0.05

No evidence that Age and Na_to_K are normal

2.5 수치형 변수 - 독립성 검정

1) Scatterplot



규칙성이 없어 보임

2) 상관계수

Correlation Coefficient = -0.0631

Age와 Na_to_K가 독립이라고 가정

2.6 범주형 변수 - 독립성 검정

Chi-Square Test of Independence

H_0 : The five variables are mutually independent

H_1 : not H_0

```
## Contingency table
from scipy import stats
cont_table = pd.crosstab(drug.Sex, [drug.BP, drug.Cholesterol, drug.Age_binned,
                                    drug.Na_to_K_binned],
rownames=["Sex"], colnames=["BP", "Cholesterol","Age_binned", "Na_to_K_binned"])
cont_table
```

```
# Chi-square test
test = stats.chi2_contingency(cont_table, correction=False)
print("Test statistic = ", test.statistic)
print("P-value = ", test.pvalue)
print("Degrees of Freedom", test.dof)
```

```
Test statistic =  84.56768925518926
P-value =  0.696338140222249
Degrees of Freedom 92
```

Cannot reject null hypothesis at α = 0.05

↓

No evidence that Sex, BP, Cholesterol, Age_binned, Na_to_K_binned are not independent

범주형 변수들끼리 독립이라고 가정 !

3

데이터 전처리

데이터 전처리

3.1 결측치 및 이상치 확인

1) 결측치 확인

```
df.isnull().sum()
```

```
Age            0
Sex            0
BP             0
Cholesterol    0
Na_to_K        0
Drug           0
dtype: int64
```

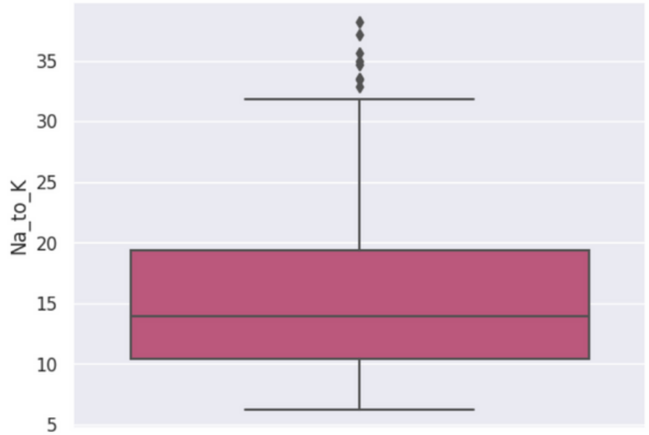
결측치 없음

2) 이상치 확인

Sex
(성별)

이상치 없음

Drug
(의약품)



이상치 8개 존재

데이터 수가 적은 관계로 이상치를 제거하지 않음

3.2 파생변수 생성

Age, Na_to_K 변수 바이닝 (Binning)

- 어떤 모델을 선택하더라도 정확도가 매우 높았음
- 과적합을 방지하기 위해 바이닝을 통해 다음의 변수를 생성

Age → Age_binned: <20s, 20s, 30s, 40s, 50s, 60s, >60s
Na_to_K → Na_to_K_binned: <10, 10-20, 20-30, >30

데이터 전처리

3.3 범주형 변수 전처리 | 변수 인코딩 (Encoding)

Label Encoding

Drug

One - Hot Encoding

Sex
BP
Cholesterol
Age_Binned
Na_to_K_Binned

3.3 범주형 변수 전처리 | 변수 인코딩 (Encoding)

```
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder

convert = preprocessing.LabelEncoder()

# 1)
drug['Drug'] = convert.fit_transform(drug['Drug'])

# 2)
one_hot = ['Sex', 'BP', 'Cholesterol', 'Age_binned', 'Na_to_K_binned']
drug = pd.get_dummies(drug, columns=one_hot)
drug.head()
```

3.4 표준화, 정규화

수치형 변수인 Age, Na_to_K에 적용

모델	표준화	정규화	이유
SVM	O	X	• 거리 기반의 학습 모델이므로 스케일링이 필요(표준화만으로도 충분)
Random Forest	X	X	• 스케일링에 민감하지 않은 모델이라 불필요
Decision Tree	X	X	• 스케일링에 민감하지 않은 모델이라 불필요
Logistic Regression	X	X	• 오차의 정규성 가정을 만족하지 않아도 됨

4

분석 모델 적용

Part 4

분석 모델 적용

1. SVM

2. Random Forest

3. Decision Tree

4. Logistic Regression

분석 모델 적용

4.1 SVM(Support Vector Machine)

1) 모델 설명

데이터를 가장 잘 나누는 초평면을 찾아내는 분류 및 회귀 알고리즘

장점

- 학습 데이터가 적은 분야도 사용 가능
- 오류 데이터의 영향이 적음

단점

- 하이퍼파라미터 조합을 통한 모형 구축에 시간이 오래 걸림

2) 하이퍼파라미터 튜닝

- grid search 사용

```
# Define the parameter grid for grid search
param_grid = {
    'C': [0.1, 1, 10],
    'kernel': ['linear', 'rbf', 'poly'],
    'degree': [2, 3, 4],
    'gamma': ['scale', 'auto', 0.1, 1]
}

svm_model = SVC()

grid_search = GridSearchCV(svm_model, param_grid, cv=3, verbose=2)

grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

y_pred = best_model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Best Parameters:", best_params)
```

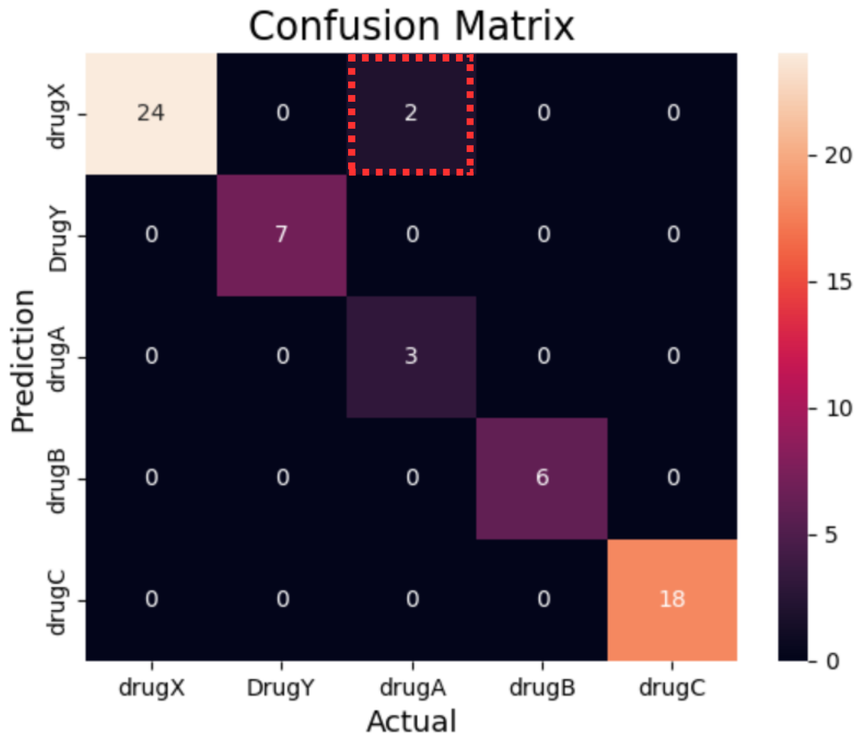
```
Best Parameters:
{'C': 10, 'degree': 2, 'gamma': 'scale', 'kernel': 'linear'}
```

분석 모델 적용

4.1 SVM(Support Vector Machine)

3) 결과

- confusion matrix



- classification report

정밀도, 재현율, F1-score, support(각 클래스의 실제 샘플 수) 지표를 제공

	precision	recall	f1-score	support
0	1.00	0.92	0.96	26
1	1.00	1.00	1.00	7
2	0.60	1.00	0.75	3
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	18
accuracy			0.97	60
macro avg	0.92	0.98	0.94	60
weighted avg	0.98	0.97	0.97	60

```
Accuracy is: 0.9666666666666667
```

분석 모델 적용

4.2 Random Forest

1) 모델 설명

결정 트리를 기본 모델로 사용하는 앙상블 방법
결정 트리를 여러 개 만들어서 그 결과들을 종합적으로 고려하여 결론을 도출하는 방법

장점

- 과적합, 이상치 및 결측치에 Robust함
- 정규화 과정이 필요 없음
- 비선형적인 데이터에도 좋은 성능을 보임

단점

- 학습 및 연산 시간이 오래 걸림
- 해석이 용이하지 않음

2) 하이퍼파라미터 튜닝

- grid search 사용

```python
from sklearn.model_selection import GridSearchCV

# 파라미터 딕셔너리 정의
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['auto', 'sqrt', 'log2']
}

# Grid Search 적용
grid_search = GridSearchCV(estimator=drug_RF, param_grid=param_grid, cv=5)

grid_search.fit(x_train, y_train)

# 최적의 파라미터 및 모델 정의
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

y_pred2_GS = best_model.predict(x_test)

accuracy = accuracy_score(y_test, y_pred2_GS)
print("Best Parameters:", best_params)
print("Accuracy:", accuracy)
```

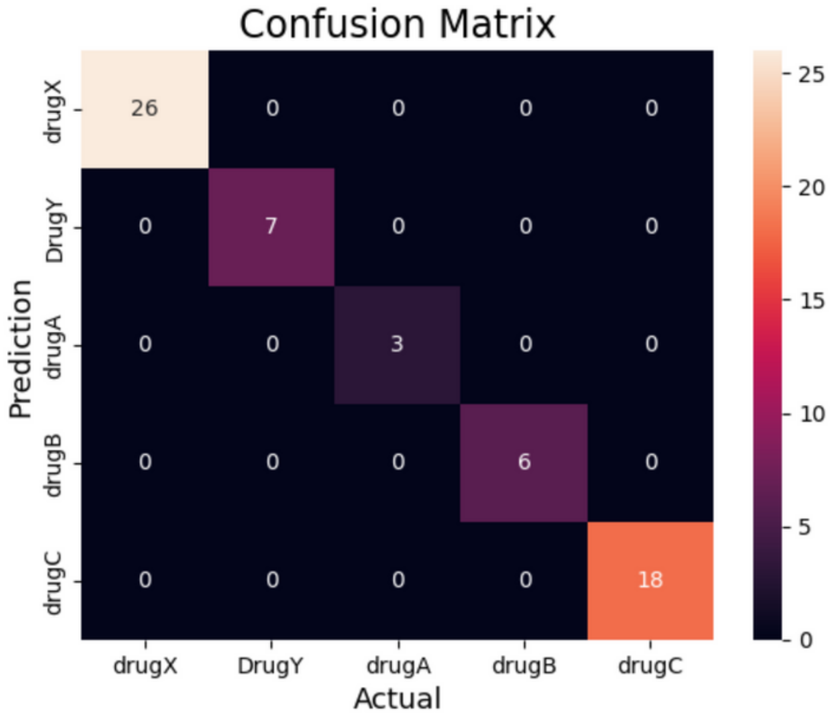
```
Best Parameters:
{'max_depth': None, 'max_features': 'auto', 'min_samples_leaf': 1,
 'min_samples_split': 2, 'n_estimators': 300, 'random_state': 42}
```

분석 모델 적용

4.2 Random Forest

3) 결과

- confusion matrix



- classification report

정밀도, 재현율, F1-score, support(각 클래스의 실제 샘플 수) 지표를 제공

	precision	recall	f1-score	support
0	1.00	1.00	1.00	26
1	1.00	1.00	1.00	7
2	1.00	1.00	1.00	3
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	18
accuracy			1.00	60
macro avg	1.00	1.00	1.00	60
weighted avg	1.00	1.00	1.00	60

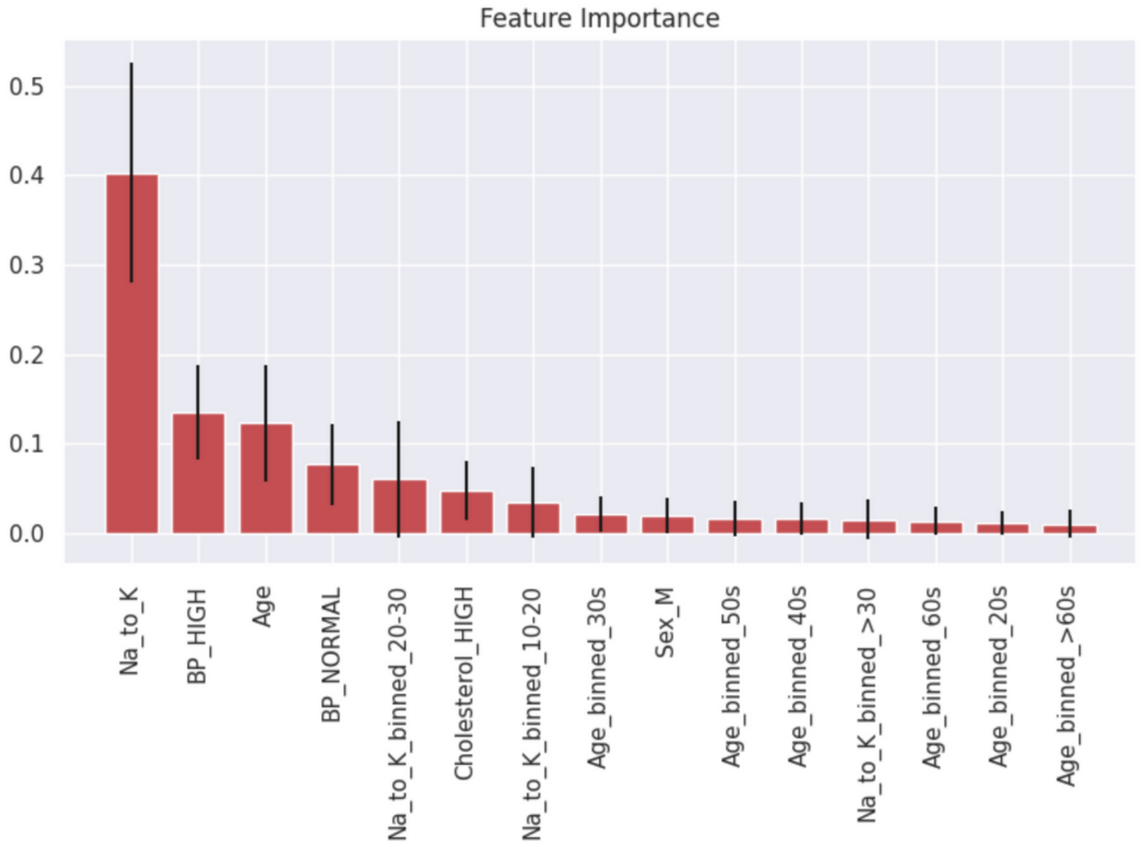
```
Accuracy: 1.0
```

분석 모델 적용

4.2 Random Forest

4) 변수 중요도 평가

```
Importance of attributes:
Na_to_K: attribute 1 (0.402701)
BP_HIGH: attribute 3 (0.135020)
Age: attribute 0 (0.122766)
BP_NORMAL: attribute 4 (0.076437)
Na_to_K_binned_20-30: attribute 13 (0.059898)
Cholesterol_HIGH: attribute 5 (0.047393)
Na_to_K_binned_10-20: attribute 12 (0.034208)
Age_binned_30s: attribute 7 (0.020654)
Sex_M: attribute 2 (0.019315)
Age_binned_50s: attribute 9 (0.016300)
Age_binned_40s: attribute 8 (0.015750)
Na_to_K_binned_>30: attribute 14 (0.014973)
Age_binned_60s: attribute 10 (0.013487)
Age_binned_20s: attribute 6 (0.010850)
Age_binned_>60s: attribute 11 (0.010246)
```



- 의약품 분류에 있어 중요하게 고려되는 변수 : Na_to_K, BP_HIGH, Age
- EDA를 통해 예측한 것처럼 Sex 변수의 중요도는 떨어짐을 확인

분석 모델 적용

4.3 Decision Tree

1) 모델 설명

데이터 학습을 통해 트리 기반의 분류 규칙을 만드는 방법

장점
- 분류 예측에 유용하며 해석이 용이함
- 데이터 전처리 불필요
- 시각화 가능

단점
- 과적합 가능성이 높음
- 데이터 수가 적은 경우 불안정

2) 하이퍼파라미터 튜닝

- grid search 사용

```
# Define the parameter grid for grid search
param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': [None, 'sqrt', 'log2']
}

dt_model = DecisionTreeClassifier()

grid_search = GridSearchCV(dt_model, param_grid, cv=3, verbose=2)

grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_
best_model = grid_search.best_estimator_

dt_pred = best_model.predict(X_test)

dt_accuracy = accuracy_score(y_test, dt_pred)
print("Best Parameters:", best_params)
```

```
Best Parameters:
{'criterion': 'gini', 'max_depth': None, 'max_features': None,
 'min_samples_leaf': 1, 'min_samples_split': 2}
```

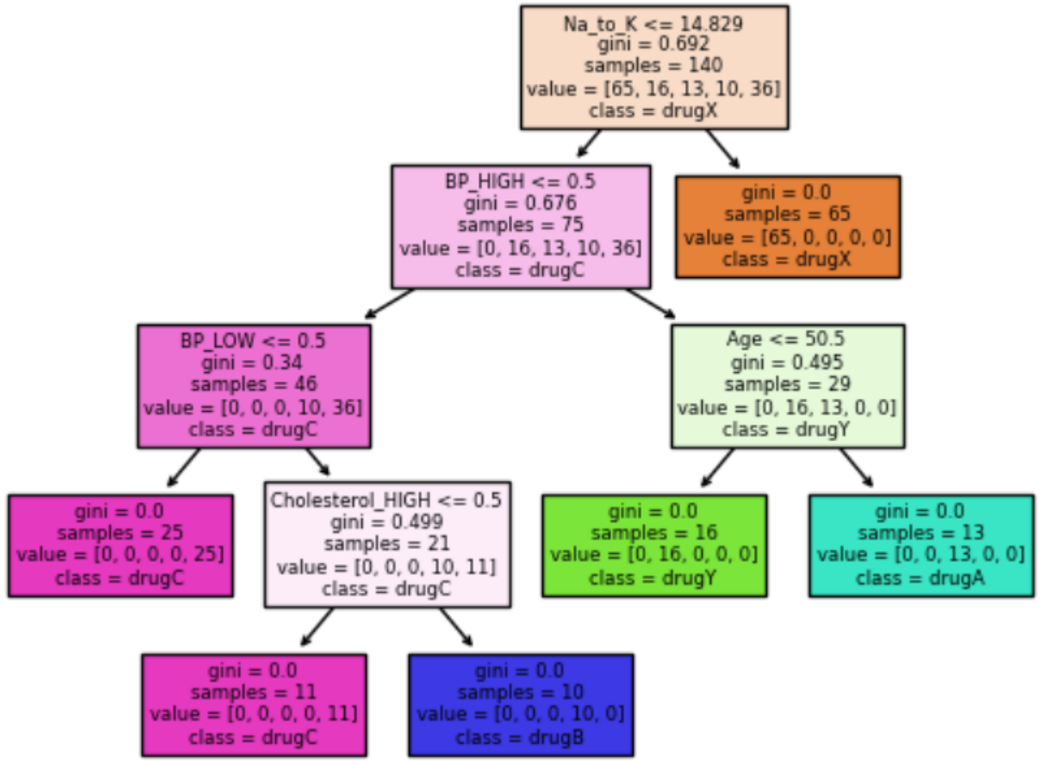
분석 모델 적용

4.3 Decision Tree

3) 결과

```
The constructed tree has  11 nodes
|--- Na_to_K <= 14.83
|   |--- BP_HIGH <= 0.50
|   |   |--- BP_LOW <= 0.50
|   |   |   |--- class: 4
|   |   |--- BP_LOW >  0.50
|   |   |   |--- Cholesterol_HIGH <= 0.50
|   |   |   |   |--- class: 4
|   |   |   |--- Cholesterol_HIGH >  0.50
|   |   |   |   |--- class: 3
|   |--- BP_HIGH >  0.50
|   |   |--- Age <= 50.50
|   |   |   |--- class: 1
|   |   |--- Age >  50.50
|   |   |   |--- class: 2
|--- Na_to_K >  14.83
|   |--- class: 0
```

- 결정트리 시각화



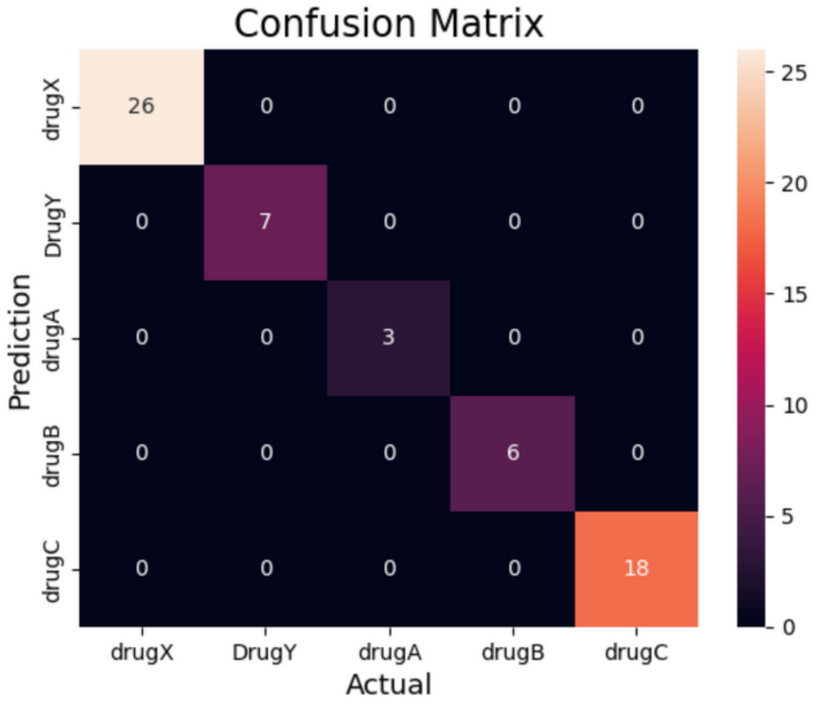
- Na_to_K, BP_HIGH 변수가 분류하는데 중요한 것으로 보임
- EDA를 통해 예측한 것처럼 Sex 변수의 중요도는 떨어짐을 확인

분석 모델 적용

4.3 Decision Tree

3) 결과

- confusion matrix



- classification report

정밀도, 재현율, F1-score, support(각 클래스의 실제 샘플 수) 지표를 제공

	precision	recall	f1-score	support
0	1.00	1.00	1.00	26
1	1.00	1.00	1.00	7
2	1.00	1.00	1.00	3
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	18
accuracy			1.00	60
macro avg	1.00	1.00	1.00	60
weighted avg	1.00	1.00	1.00	60

```
Accuracy is: 1.0
```

분석 모델 적용

4.4 Multiclass Logistic Regression

1) 모델 설명

선형 회귀 방식을 Classification 문제에 적용한 방법

장점

- 회귀계수 (Coefficient)가 계산되며, 해석이 용이함
- 클래스에 속할 확률을 구할 수 있다.

단점

- 선형 모형이다.
- 상호작용이 있을 경우 수동으로 변수를 추가해야 함.

2) 하이퍼파라미터 튜닝

- 별도의 튜닝 과정 없음
- Baseline column을 제거해야 한다는 점 주의 (가장 작은 범주를 제거)
  - Sex: Female 제거
  - BP: BP_LOW 제거
  - Cholesterol: Cholesterol_NORMAL 제거
  - Age_binned: Age_binned_<20s 제거
  - Na_to_K_binned: Na_to_K_binned_<10 제거

4.4 Multiclass Logistic Regression

3) 결과

- Coefficients

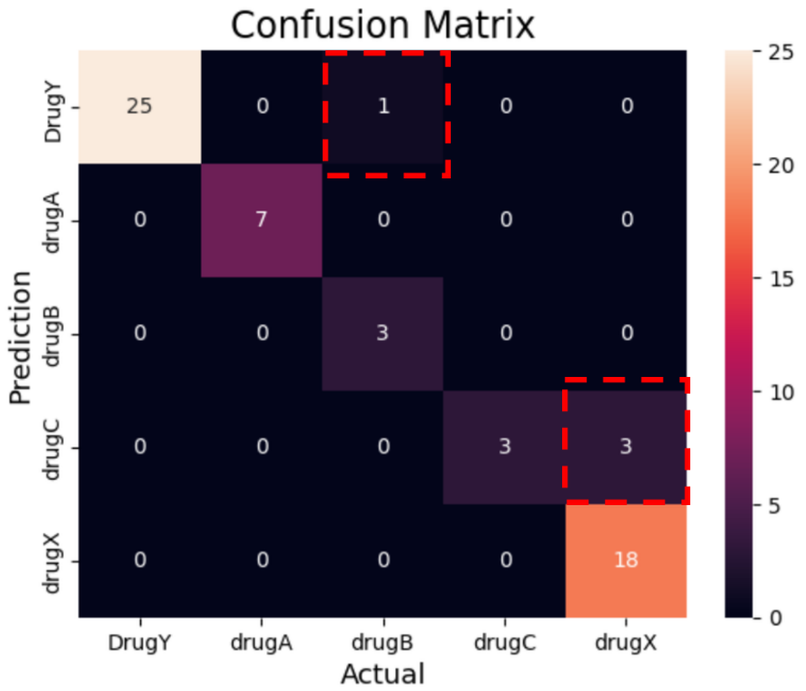
Age	Na_to_K	Sex_M	BP_HIGH	BP_NORMAL	Cholesterol_HIGH	Age_binned_20s	Age_binned_30s	Age_binned_40s	Age_binned_50s	Age_binned_60s	Age_binned_>60s	Na_to_K_binned_10-20	Na_to_K_binned_20-30	Na_to_K_binned_>30	
drugY	-0.024079	2.050701	0.165550	-0.322051	0.188647	0.075999	-0.094422	-0.018406	0.112359	-0.077269	0.241947	-0.173754	-0.002503	0.000048	-2.794305e-06
drugA	-0.082794	-0.607678	-0.003054	2.126726	-0.506448	0.137994	-0.068678	0.455941	0.508993	-0.132711	-0.298340	-0.052269	0.017794	0.000017	1.587805e-06
drugB	0.133053	-0.342913	-0.012823	1.743605	-0.461141	-0.347242	-0.011853	-0.126747	-0.456707	0.745794	0.091563	-0.239383	0.114461	-0.000052	2.297349e-07
drugC	-0.018430	-0.598324	0.239900	-1.506994	-1.103936	1.382675	0.209649	-0.520925	0.184552	-0.500932	0.000703	0.293329	0.058000	0.000007	2.703918e-07
drugX	-0.007750	-0.501785	-0.389573	-2.041286	1.882878	-1.249426	-0.034696	0.210136	-0.349198	-0.034882	-0.035873	0.172076	-0.187752	-0.000020	7.063729e-07

분석 모델 적용

4.4 Multiclass Logistic Regression

3) 결과

- confusion matrix



- classification report

	precision	recall	f1-score	support
0	1.00	0.96	0.98	26
1	1.00	1.00	1.00	7
2	0.75	1.00	0.86	3
3	1.00	0.50	0.67	6
4	0.86	1.00	0.92	18
accuracy			0.93	60
macro avg	0.92	0.89	0.89	60
weighted avg	0.94	0.93	0.93	60

```
Accuracy is:  0.9333333333333333
```

5

결론 및 고찰

결론 및 고찰

5.1 분석 결과 도출

SVM

Accuracy : 0.967

Random forest

Accuracy : 1.0

Decision tree

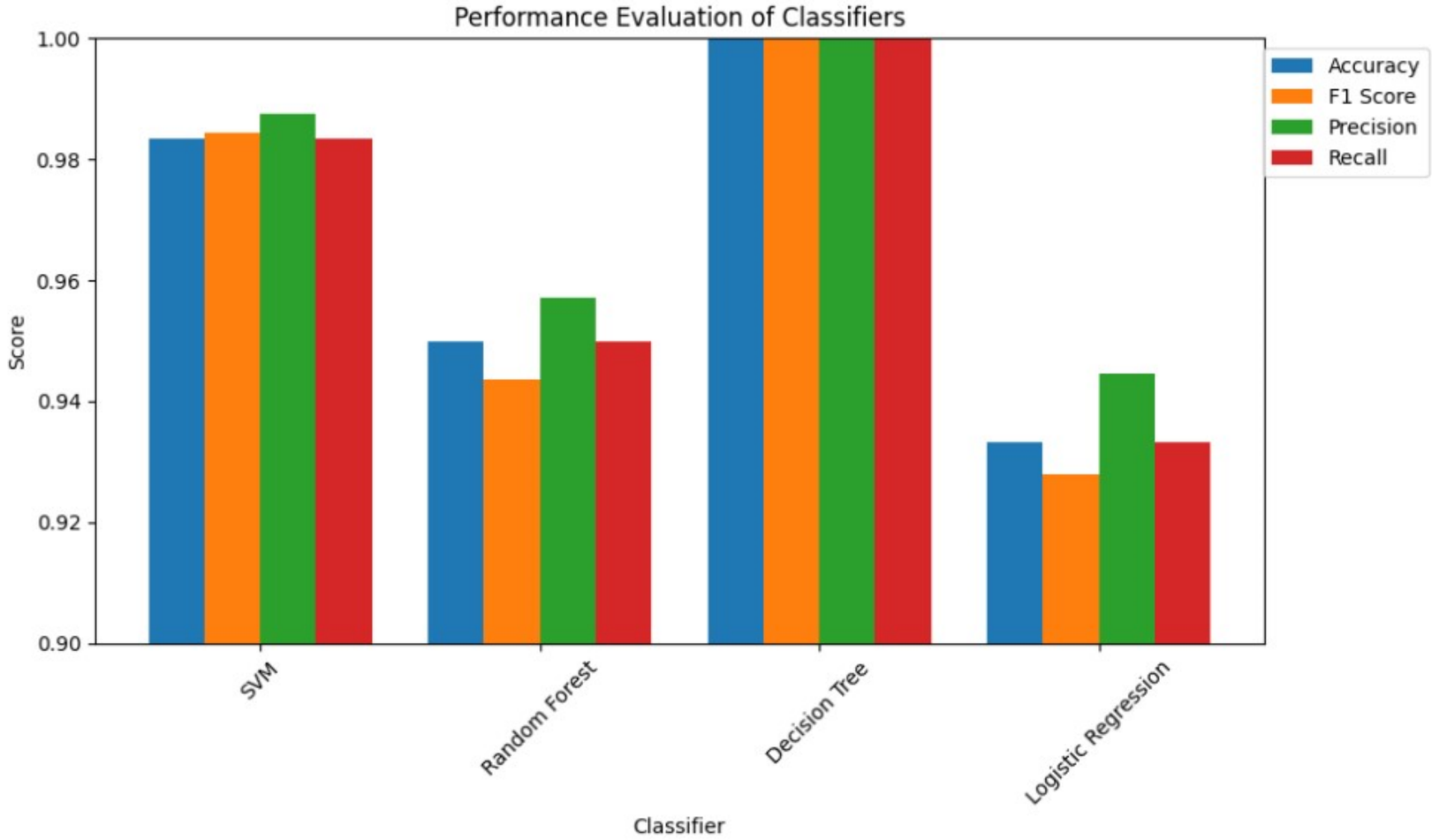
Accuracy : 1.0

Logistic Regression

Accuracy : 0.933

결론 및 고찰

5.1 분석 결과 도출



Performance: Decision tree > SVM > Random forest > Logistic Regression

결론 및 고찰

5.2 프로젝트 결론

- 모델 성능: 4개의 모델 모두 성능이 높게 나왔음(0.9 이상)

- 하지만 관측값의 개수가 200개에 불과하기 때문에, 과적합이 발생했을 가능성이 존재함
  -> 더 많은 양의 데이터로 모델링을 진행할 경우 더 정확한 결과를 얻을 수 있다고 판단됨

- 변수 중요도: Drug를 분류하는 데에 있어서 Na_to_K, BP, Age 순서로 높은 중요도를 가지는 것으로 판단됨

- EDA 과정에서 SEX 변수가 Drug를 분류에 있어 영향을 미치지 않는 것이 보여졌는데, 실제로 랜덤포레스트, 결정트리 결과에서 SEX 변수가 낮은 중요도를 갖는 것을 확인할 수 있었음

5.3 프로젝트를 마치며

프로젝트 의의

- EDA부터 분석 결과까지 모델링의 전 단계를 직접 구현하는 과정이 매우 유익했음
- 다양한 EDA 과정을 거치며 변수 간의 관계성을 알아가는 과정이 인상 깊었음
- 모델별로 적합한 전처리 과정을 고민하고, 실제로도 모델마다 각기 다른 전처리 과정을 적용하는 과정에서 많은 부분들을 알아갈 수 있었음
- 트리 계열 모델과 SVM, 로지스틱 회귀 등 다양한 분류 모델을 탐구하고 적용하는 과정이 뜻 깊었음

프로젝트 한계점

- 어떤 모델을 쓰더라도 결과가 잘 나와, 최적의 모델을 판단하기는 어려움
- 높은 prediction accuracy가 과적합에 의한 것인지, 데이터가 명확했기 때문인지 판단하기 어려음
- 모델 선정 시 수치형 변수를 (Age. Na_to_K) 제외할 경우 prediction accuracy가 매우 떨어짐
- 다중분류이기 때문에 ROC curve는 사용하지 않음
- Naive Bayes, KNN, Neural Net 디벨롭 여지가 있음

감 사 합 니 다